

# Namespace AlohaKit.Controls

## Classes

### [Avatar](#)

The Avatar is a custom drawn control that provides a flexible and efficient way to render avatar-like graphics, such as profile pictures, icons, or other visual representations, in an application.

### [AvatarDrawable](#)

### [AvatarSizeExtensions](#)

The AvatarSizeExtensions class provides extension methods for the AvatarSize enumeration, allowing developers to retrieve various sizing details for avatars based on their size (e.g., dimensions, indicator sizes, font sizes).

### [BarChart](#)

The BarChart is a drawn control used to render bar charts, allowing for the visualization of data as rectangular bars. Each bar's length or height corresponds to the value it represents.

It extends the BaseChart class, providing additional functionality tailored to bar chart rendering while leveraging shared chart capabilities.

### [BarChartDrawable](#)

### [BaseChart](#)

### [BaseChartDrawable](#)

### [BusyIndicator](#)

The BusyIndicator is a drawn control that provides a graphical representation to indicate an application or process is busy.

### [BusyIndicatorDrawable](#)

### [Button](#)

The Button represents a customizable drawn button control. It allows developers to create interactive buttons with highly customizable designs, leveraging the power of lightweight graphics rendering.

### [ButtonDrawable](#)

### [Captcha](#)

The Captcha is a drawn control used to render and manage CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenges. It leverages .NET MAUI Graphics for efficient graphical rendering and provides a flexible way to generate and validate CAPTCHAs in an application.

## [CaptchaDrawable](#)

### [CheckBox](#)

The CheckBox is a custom drawn control that represents a checkbox for selecting or deselecting options. It is highly customizable while maintaining efficient rendering capabilities.

### [CheckBoxDrawable](#)

### [LineChart](#)

The LineChart provides a drawn control to visualize data as a line chart. It allows data points to be represented by a continuous line, making it ideal for showing trends over time or relationships between variables.

This class inherits from BaseChart, leveraging shared charting functionality while adding line-specific rendering features.

### [LineChartDrawable](#)

### [LinearGauge](#)

The LinearGauge is a drawn control for rendering a linear gauge. A linear gauge is a visual representation of data along a straight or horizontal/vertical axis, commonly used to display values such as progress, measurements, or performance indicators.

### [LinearGaugeDrawable](#)

### [MultiBarChart](#)

The MultiBarChart is a drawn control designed to display multiple bar charts within a single charting view. It enables the visualization of grouped or stacked data, making it an ideal tool for comparing multiple datasets across categories.

This class inherits from BaseChart, leveraging core charting functionality while adding features specific to bar chart rendering.

### [MultiBarChartDrawable](#)

### [MultiLineChartDrawable](#)

### [MultiLineChartView](#)

The MultiLineChartView is a drawn control for displaying multiple line charts within a single view. This enables the visualization of multiple data series, each represented by a separate line, on a shared coordinate system.

The class is derived from BaseChart, allowing it to inherit essential charting properties and behaviors while adding specialized features for multi-line visualization.

## [NumericUpDown](#)

The NumericUpDown is a drawn control for selecting numeric values by incrementing or decrementing them with interactive buttons or input.

## [NumericUpDownDrawable](#)

### [PieChart](#)

The PieChart is a drawn control for rendering pie charts to visually represent data as portions of a circle.

## [PieChartDrawable](#)

### [ProgressBar](#)

The ProgressBar class represents a customizable control for displaying the progress of a task visually.

## [ProgressBarDrawable](#)

### [ProgressRadial](#)

The ProgressRadial is a drawn control designed to visually represent progress in a circular or radial form. It can be used to display percentages or completion statuses, such as task progress, loading indicators, or performance metrics.

## [ProgressRadialDrawable](#)

### [PulseIcon](#)

The PulseIcon is a drawn control that creates an animated pulsing effect around an icon. This control is particularly useful for drawing attention to specific elements in your user interface, such as notifications, interactive features, or status indicators.

## [PulseIconDrawable](#)

### [Rating](#)

The Rating is a drawn control that allows users to rate items or entities, typically using a star-based system or other graphical representation. This control is perfect for scenarios where users need to provide feedback, score performance, or indicate preferences in an intuitive visual format.

## [RatingDrawable](#)

### [RatingValueChangedEventArgs](#)

The RatingValueChangedEventArgs class provides data for events that report a change in the rating value of a Rating control.

## [SegmentedControl](#)

The SegmentedControl is a drawn control that allows users to make a single selection from a set of segmented options. It is ideal for scenarios where a user needs to choose between mutually exclusive

options, such as switching between tabs or filtering content.

## [SegmentedControlDrawable](#)

### [SelectedIndexEventArgs](#)

The SelectedIndexEventArgs class provides data for events that report a change in the selected index of a SegmentedControl.

## [Slider](#)

The Slider class is a graphical control that allows users to select a value from a continuous or discrete range by dragging a thumb along a track. This versatile control is widely used for adjusting settings like volume, brightness, or custom application parameters.

## [SliderDrawable](#)

### [ToggleSwitch](#)

The ToggleSwitch is a drawn control that allows users to toggle between two states (on and off). Designed for visual clarity and smooth interactivity, this control is ideal for enabling or disabling settings, features, or options within an application.

## [ToggleSwitchDrawable](#)

# Enums

### [AvatarSize](#)

The AvatarSize enumeration represents predefined sizes for an avatar, providing a standardized way to specify dimensions across an application.

### [CaptchaLevel](#)

The CaptchaLevel enumeration defines the different levels of complexity for CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenges. This can be used to configure or determine the strength of a CAPTCHA system, balancing between ease of use and security requirements.

### [ProgressBarStyle](#)

The ProgressBarStyle enumeration defines the visual style options for rendering a progress bar. This provides developers with a simple way to specify whether the progress bar appears with sharp corners or rounded edges.

### [ProgressRadialDirection](#)

The ProgressRadialDirection enumeration specifies the direction of progress rendering for radial progress components, allowing developers to define whether progress moves clockwise or counterclockwise along the circular axis.

## ThumbShape

The ThumbShape enumeration defines the possible shapes for a "thumb" in UI components, such as sliders, progress indicators, or drag handles. It provides developers with options to customize the appearance of the thumb based on design requirements.

# Class Avatar

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Avatar is a custom drawn control that provides a flexible and efficient way to render avatar-like graphics, such as profile pictures, icons, or other visual representations, in an application.

```
public class Avatar : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← Avatar

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),

[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,

[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### Avatar()

```
public Avatar()
```

## Fields

### AvatarSizeProperty

```
public static readonly BindableProperty AvatarSizeProperty
```

## Field Value

[BindableProperty](#)

## BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

Field Value

[BindableProperty](#) ↗

## FillProperty

```
public static readonly BindableProperty FillProperty
```

Field Value

[BindableProperty](#) ↗

## NameProperty

```
public static readonly BindableProperty NameProperty
```

Field Value

[BindableProperty](#) ↗

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty](#) ↗

# Properties

## AvatarSize

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

### Property Value

[AvatarSize](#)

## Background

```
public Brush Background { get; set; }
```

### Property Value

[Brush](#)

## Fill

```
public Brush Fill { get; set; }
```

### Property Value

[Brush](#)

## Name

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

### Property Value

[string](#)

# PersonaDrawable

```
public AvatarDrawable PersonaDrawable { get; set; }
```

## Property Value

[AvatarDrawable](#)

# TextColor

```
public Color TextColor { get; set; }
```

## Property Value

[Color](#)

# Methods

## OnParentSet()

Raises the (internal) [ParentSet](#) event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet](#)()

# Class AvatarDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class AvatarDrawable : IDrawable
```

## Inheritance

[object](#) ← AvatarDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## FillPaint

```
public Paint FillPaint { get; set; }
```

### Property Value

[Paint](#)

## FontSize

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

### Property Value

[double](#)

## Text

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

### Property Value

[string](#)

## TextColor

```
public Color TextColor { get; set; }
```

### Property Value

[Color](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawFill(ICanvas, RectF)

```
public virtual void DrawFill(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawInitials(ICanvas, RectF)

```
public virtual void DrawInitials(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Enum AvatarSize

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The AvatarSize enumeration represents predefined sizes for an avatar, providing a standardized way to specify dimensions across an application.

```
public enum AvatarSize
```

## Extension Methods

[AvatarSizeExtensions.GetAvatarIndicatorIconScale\(AvatarSize\)](#),

[AvatarSizeExtensions.GetAvatarIndicatorSize\(AvatarSize\)](#),

[AvatarSizeExtensions.GetAvatarSize\(AvatarSize\)](#), [AvatarSizeExtensions.GetInitialsFontSize\(AvatarSize\)](#)

## Fields

Large = 1

Small = 0

XXLarge = 2

# Class AvatarSizeExtensions

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The AvatarSizeExtensions class provides extension methods for the AvatarSize enumeration, allowing developers to retrieve various sizing details for avatars based on their size (e.g., dimensions, indicator sizes, font sizes).

```
public static class AvatarSizeExtensions
```

## Inheritance

[object](#) ← AvatarSizeExtensions

## Inherited Members

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

## Methods

### GetAvatarIndicatorIconScale(AvatarSize)

```
public static float GetAvatarIndicatorIconScale(this AvatarSize avatarSize)
```

#### Parameters

avatarSize [AvatarSize](#)

#### Returns

[float](#)

### GetAvatarIndicatorSize(AvatarSize)

```
public static int GetAvatarIndicatorSize(this AvatarSize avatarSize)
```

Parameters

avatarSize [AvatarSize](#)

Returns

[int](#)

## GetAvatarSize(AvatarSize)

```
public static int GetAvatarSize(this AvatarSize avatarSize)
```

Parameters

avatarSize [AvatarSize](#)

Returns

[int](#)

## GetInitialsFontSize(AvatarSize)

```
public static int GetInitialsFontSize(this AvatarSize avatarSize)
```

Parameters

avatarSize [AvatarSize](#)

Returns

[int](#)

# Class BarChart

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The BarChart is a drawn control used to render bar charts, allowing for the visualization of data as rectangular bars. Each bar's length or height corresponds to the value it represents.

It extends the BaseChart class, providing additional functionality tailored to bar chart rendering while leveraging shared chart capabilities.

```
public sealed class BarChart : BaseChart, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform, IDisposable
```

## Inheritance

[Object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← [BaseChart](#) ← BarChart

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#),  
[IDisposable](#)

## Inherited Members

[BaseChart.EasingProperty](#), [BaseChart.Easing](#), [BaseChart.BackgroundProperty](#), [BaseChart.Background](#),  
[BaseChart.AxisDashPatternProperty](#), [BaseChart.AxisDashPattern](#),  
[BaseChart.DisplayHeaderValuesProperty](#), [BaseChart.DisplayHeaderValues](#),  
[BaseChart.PathsColorOpacityProperty](#), [BaseChart.PathsColorOpacity](#),  
[BaseChart.AxisLinesFontSizeProperty](#), [BaseChart.AxisLinesFontSize](#), [BaseChart.AxisLineColorProperty](#),  
[BaseChart.AxisLinesColor](#), [BaseChart.AxisLinesStrokeSizeProperty](#), [BaseChart.AxisLinesStrokeSize](#),  
[BaseChart.DisplayVerticalAxisLinesProperty](#), [BaseChart.DisplayVerticalAxisLines](#),  
[BaseChart.DisplayHorizontalAxisLinesProperty](#), [BaseChart.DisplayHorizontalAxisLines](#),  
[BaseChart.AnimationIntervalProperty](#), [BaseChart.AnimationInterval](#),  
[BaseChart.ReanimateOnPropertyChangedProperty](#), [BaseChart.ReanimateOnPropertyChanged](#),  
[BaseChart.EnableEntryAnimationsProperty](#), [BaseChart.EnableEntryAnimations](#),

[BaseChart.EntriesProperty](#), [BaseChart.Entries](#), [BaseChart.IsLabelTextTruncationEnabledProperty](#),  
[BaseChart.IsLabelTextTruncationEnabled](#), [BaseChart.DisplayValueLabelsOnTopProperty](#),  
[BaseChart.DisplayValueLabelsOnTop](#), [BaseChart.StrokeSizeProperty](#), [BaseChart.StrokeSize](#),  
[BaseChart.StrokeColorProperty](#), [BaseChart.StrokeColor](#), [BaseChart.ItemSeparationMarginProperty](#),  
[BaseChart.ItemSeparationMargin](#), [BaseChart.FooterLabelsTextSizeProperty](#),  
[BaseChart.FooterLabelsTextSize](#), [BaseChart.FooterLabelsMarginProperty](#),  
[BaseChart.FooterLabelsMargin](#), [BaseChart.HeaderValuesMarginProperty](#),  
[BaseChart.HeaderValuesMargin](#), [BaseChart.FillColorProperty](#), [BaseChart.FillColor](#),  
[BaseChart.FontFamilyProperty](#), [BaseChart.FontFamily](#), [BaseChart.FontSizeProperty](#), [BaseChart.FontSize](#),  
[BaseChart.ChartMarginProperty](#), [BaseChart.ChartMargin](#), [BaseChart.FontColorProperty](#),  
[BaseChart.FontColor](#), [BaseChart.FooterLabelsFontColorProperty](#), [BaseChart.FooterLabelsFontColor](#),  
[BaseChart.EnableAntialiasProperty](#), [BaseChart.EnableAntialias](#), [BaseChart.Dispose\(\)](#),  
[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.GetChildElements\(Point\)](#),  
[View.GestureRecognizers](#), [View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#),  
[VisualElement.NavigationProperty](#), [VisualElement.StyleProperty](#),  
[VisualElement.InputTransparentProperty](#), [VisualElement.IsEnabledProperty](#),  
[VisualElement.XProperty](#), [VisualElement.YProperty](#), [VisualElement.AnchorXProperty](#),  
[VisualElement.AnchorYProperty](#), [VisualElement.TranslationXProperty](#),  
[VisualElement.TranslationYProperty](#), [VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#),  
[VisualElement.RotationProperty](#), [VisualElement.RotationXProperty](#),  
[VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#), [VisualElement.ScaleXProperty](#),  
[VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#), [VisualElement.VisualProperty](#),  
[VisualElement.IsVisibleProperty](#), [VisualElement.OpacityProperty](#),  
[VisualElement.BackgroundColorProperty](#), [VisualElement.BehaviorsProperty](#),  
[VisualElement.TriggersProperty](#), [VisualElement.WidthRequestProperty](#),  
[VisualElement.HeightRequestProperty](#), [VisualElement.MinimumWidthRequestProperty](#),  
[VisualElement.MinimumHeightRequestProperty](#), [VisualElement.MaximumWidthRequestProperty](#),  
[VisualElement.MaximumHeightRequestProperty](#), [VisualElement.IsFocusedProperty](#),  
[VisualElement.FlowDirectionProperty](#), [VisualElement.WindowProperty](#),  
[VisualElement.ShadowProperty](#), [VisualElement.ZIndexProperty](#),  
[VisualElement.ControlsVisualElementMapper](#), [VisualElement.BatchBegin\(\)](#),  
[VisualElement.BatchCommit\(\)](#), [VisualElement.Focus\(\)](#),  
[VisualElement.Measure\(double, double, MeasureFlags\)](#), [VisualElement.Unfocus\(\)](#),  
[VisualElement.Arrange\(Rect\)](#), [VisualElement.Layout\(Rect\)](#),  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#),  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#), [VisualElement.Visual](#),

[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsFocused](#) ,  
[VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## BarChart()

```
public BarChart()
```

# Fields

## BackgroundBarsFillColorProperty

```
public static readonly BindableProperty BackgroundBarsFillColorProperty
```

Field Value

[BindableProperty](#)

## BarsFillColorProperty

```
public static readonly BindableProperty BarsFillColorProperty
```

Field Value

[BindableProperty](#)

## ShowBackgroundBarsProperty

```
public static readonly BindableProperty ShowBackgroundBarsProperty
```

Field Value

[BindableProperty](#)

# Properties

## BackgroundBarsFillColor

Gets or sets the background color to use when drawing each bar

```
public Color BackgroundBarsFillColor { get; set; }
```

Property Value

[Color](#) ↗

## BarsFillColor

Gets or sets the color to use when drawing each bar

```
public Color BarsFillColor { get; set; }
```

Property Value

[Color](#) ↗

## ShowBackgroundBars

If true chart will draw background bars and value bars. If not only value bars will be drawn. Default is true

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

Property Value

[bool](#) ↗

# Class BarChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public sealed class BarChartDrawable : BaseChartDrawable, IDrawable, IDisposable
```

## Inheritance

[object](#) ← [BaseChartDrawable](#) ← BarChartDrawable

## Implements

[IDisposable](#), [IDisposable](#)

## Inherited Members

[BaseChartDrawable.Initialized](#), [BaseChartDrawable.IsAnimating](#), [BaseChartDrawable.ColorBrush](#),  
[BaseChartDrawable.AxisDashPattern](#), [BaseChartDrawable.DisplayHeaderValues](#),  
[BaseChartDrawable.PathsColorOpacity](#), [BaseChartDrawable.DisplayHorizontalAxisLines](#),  
[BaseChartDrawable.DisplayVerticalAxisLines](#), [BaseChartDrawable.AnimationProgress](#),  
[BaseChartDrawable.LabelTextTruncationEnabled](#), [BaseChartDrawable.DisplayValueLabelsOnTop](#),  
[BaseChartDrawable.StrokeSize](#), [BaseChartDrawable.AxisLinesStrokeSize](#),  
[BaseChartDrawable.AxisLinesColor](#), [BaseChartDrawable.StrokeColor](#),  
[BaseChartDrawable.ItemSeparationMargin](#), [BaseChartDrawable.FooterLabelsTextSize](#),  
[BaseChartDrawable.FooterLabelsMargin](#), [BaseChartDrawable.HeaderValuesMargin](#),  
[BaseChartDrawable.FillColor](#), [BaseChartDrawable.Font](#), [BaseChartDrawable.Entries](#),  
[BaseChartDrawable.FontSize](#), [BaseChartDrawable.AxisFontSize](#), [BaseChartDrawable.Margin](#),  
[BaseChartDrawable.FontColor](#), [BaseChartDrawable.FooterLabelsFontColor](#),  
[BaseChartDrawable.EnableAntialias](#), [BaseChartDrawable.Draw\(ICanvas, RectF\)](#),  
[BaseChartDrawable.ClearBackground\(ICanvas, SizeF, Color\)](#), [BaseChartDrawable.Dispose\(\)](#),  
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Properties

### BackgroundBarsFillColor

Gets or sets the background color to use when drawing each bar

```
public Color BackgroundBarsFillColor { get; set; }
```

Property Value

[Color](#)

## BarsFillColor

Gets or sets the color to use when drawing each bar

```
public Color BarsFillColor { get; set; }
```

Property Value

[Color](#)

## ShowBackgroundBars

If true chart will draw background bars and value bars. If not only value bars will be drawn. Default is true

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

Property Value

[bool](#)

# Methods

## DrawChart(ICanvas, RectF)

Calculates and draws the current chart with the Entries[] data

```
public override void DrawChart(ICanvas canvas, RectF dirtyRect)
```

Parameters

**canvas** [ICanvas](#)

Current canvas

`dirtyRect` [RectF](#)

Current canvas rect

## DrawVerticalStepLines(ICanvas, PointF[], float, float)

```
protected override void DrawVerticalStepLines(ICanvas canvas, PointF[] points, float origin,  
float maxPositiveValue)
```

### Parameters

`canvas` [ICanvas](#)

`points` [PointF](#)[]

`origin` [float](#)

`maxPositiveValue` [float](#)

# Class BaseChart

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public abstract class BaseChart : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform, IDisposable
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← BaseChart

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#),  
[IDisposable](#)

## Derived

[BarChart](#), [LineChart](#), [MultiBarChart](#), [MultiLineChartView](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.StartHoverInteraction](#),  
[GraphicsView.MoveHoverInteraction](#), [GraphicsView.EndHoverInteraction](#),  
[GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#), [GraphicsView.EndInteraction](#),  
[GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#), [View.HorizontalOptionsProperty](#),  
[View.MarginProperty](#), [View.propertyMapper](#), [View.ChangeVisualState\(\)](#),  
[View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.OnParentSet\(\)](#) , [NavigableElement.Navigation](#) , [NavigableElement.Style](#) ,  
[NavigableElement.StyleClass](#) , [NavigableElement.class](#) , [Element.AutomationIdProperty](#) ,  
[Element.ClassIdProperty](#) , [Element.ControlsElementMapper](#) ,  
[Element.InsertLogicalChild\(int, Element\)](#) , [Element.AddLogicalChild\(Element\)](#) ,  
[Element.RemoveLogicalChild\(Element\)](#) , [Element.ClearLogicalChildren\(\)](#) ,

[Element.FindByName\(string\)](#) , [Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### BaseChart()

```
public BaseChart()
```

## Fields

### AnimationIntervalProperty

```
public static readonly BindableProperty AnimationIntervalProperty
```

## Field Value

[BindableProperty](#)

## AxisDashPatternProperty

```
public static readonly BindableProperty AxisDashPatternProperty
```

Field Value

[BindableProperty](#)

## AxisLineColorProperty

```
public static readonly BindableProperty AxisLineColorProperty
```

Field Value

[BindableProperty](#)

## AxisLinesFontSizeProperty

```
public static readonly BindableProperty AxisLinesFontSizeProperty
```

Field Value

[BindableProperty](#)

## AxisLinesStrokeSizeProperty

```
public static readonly BindableProperty AxisLinesStrokeSizeProperty
```

Field Value

[BindableProperty](#)

## BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

Field Value

[BindableProperty ↗](#)

## ChartMarginProperty

```
public static readonly BindableProperty ChartMarginProperty
```

Field Value

[BindableProperty ↗](#)

## DisplayHeaderValuesProperty

```
public static readonly BindableProperty DisplayHeaderValuesProperty
```

Field Value

[BindableProperty ↗](#)

## DisplayHorizontalAxisLinesProperty

```
public static readonly BindableProperty DisplayHorizontalAxisLinesProperty
```

Field Value

[BindableProperty ↗](#)

## DisplayValueLabelsOnTopProperty

```
public static readonly BindableProperty DisplayNameLabelsOnTopProperty
```

Field Value

[BindableProperty](#)

## DisplayVerticalAxisLinesProperty

```
public static readonly BindableProperty DisplayVerticalAxisLinesProperty
```

Field Value

[BindableProperty](#)

## EasingProperty

```
public static readonly BindableProperty EasingProperty
```

Field Value

[BindableProperty](#)

## EnableAntialiasProperty

```
public static readonly BindableProperty EnableAntialiasProperty
```

Field Value

[BindableProperty](#)

## EnableEntryAnimationsProperty

```
public static readonly BindableProperty EnableEntryAnimationsProperty
```

Field Value

[BindableProperty](#)

## EntriesProperty

```
public static readonly BindableProperty EntriesProperty
```

Field Value

[BindableProperty](#)

## FillColorProperty

```
public static readonly BindableProperty FillColorProperty
```

Field Value

[BindableProperty](#)

## FontColorProperty

```
public static readonly BindableProperty FontColorProperty
```

Field Value

[BindableProperty](#)

## FontFamilyProperty

```
public static readonly BindableProperty FontFamilyProperty
```

Field Value

[BindableProperty](#)

## FontSizeProperty

```
public static readonly BindableProperty FontSizeProperty
```

Field Value

[BindableProperty](#)

## FooterLabelsFontColorProperty

```
public static readonly BindableProperty FooterLabelsFontColorProperty
```

Field Value

[BindableProperty](#)

## FooterLabelsMarginProperty

```
public static readonly BindableProperty FooterLabelsMarginProperty
```

Field Value

[BindableProperty](#)

## FooterLabelTextSizeProperty

```
public static readonly BindableProperty FooterLabelsFontSizeProperty
```

Field Value

[BindableProperty](#)

## HeaderValuesMarginProperty

```
public static readonly BindableProperty HeaderValuesMarginProperty
```

Field Value

[BindableProperty](#)

## LabelTextTruncationEnabledProperty

```
public static readonly BindableProperty IsLabelTextTruncationEnabledProperty
```

Field Value

[BindableProperty](#)

## ItemSeparationMarginProperty

```
public static readonly BindableProperty ItemSeparationMarginProperty
```

Field Value

[BindableProperty](#)

## PathsColorOpacityProperty

```
public static readonly BindableProperty PathsColorOpacityProperty
```

Field Value

[BindableProperty](#)

## ReanimateOnPropertyChangedProperty

```
public static readonly BindableProperty ReanimateOnPropertyChangedProperty
```

Field Value

[BindableProperty](#)

## StrokeColorProperty

```
public static readonly BindableProperty StrokeColorProperty
```

Field Value

[BindableProperty](#)

## StrokeSizeProperty

```
public static readonly BindableProperty StrokeSizeProperty
```

Field Value

[BindableProperty](#)

# Properties

## AnimationInterval

Sets the interval in ms to use when triggering animations. Default is 1250

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

Property Value

[int](#)

## AxisDashPattern

```
public float[] AxisDashPattern { get; set; }
```

Property Value

[float](#)[]

## AxisLinesColor

Axis lines color. Default is Lightgray

```
public Color AxisLinesColor { get; set; }
```

Property Value

[Color](#)

## AxisLinesFontSize

Axis lines font size. Default is 11

```
public float AxisLinesFontSize { get; set; }
```

Property Value

[float](#)

## AxisLinesStrokeSize

Axis lines stroke size. Default is 0.9

```
public float AxisLinesStrokeSize { get; set; }
```

Property Value

[float](#)

## Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

## ChartMargin

Gets or sets the default top and bottom internal margin to use when drawing canvas content. Default value is 15

```
public float ChartMargin { get; set; }
```

Property Value

[float](#)

## DisplayHeaderValues

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

Property Value

[bool](#)

## DisplayHorizontalAxisLines

If true header labels will be hidden and chart will draw horizontal step lines.

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

Property Value

[bool](#)

## DisplayValueLabelsOnTop

If true, chart will show value labels on top of canvas. Default is true

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

Property Value

[bool](#)

## DisplayVerticalAxisLines

If true vertical lines will be drawn as background along with horizontal lines. DisplayHorizontalAxisLines prop needs to be true as well

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

Property Value

[bool](#)

## Drawable

Gets or sets current IDrawable surface. This property MUST be assigned before ChartEntries prop

```
protected BaseChartDrawable Drawable { get; set; }
```

Property Value

[BaseChartDrawable](#)

## Easing

Sets the Easing to use when triggering animations. Default is BounceOut

```
public Easing Easing { get; set; }
```

Property Value

[Easing](#)

## EnableAntialias

Gets or sets if the current canvas will use Antialias mode or not when drawing content. Default is true

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

Property Value

[bool](#)

## EnableEntryAnimations

If true, chart will trigger animations when drawing chart content. Default is true

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

Property Value

[bool](#)

## Entries

```
public ObservableCollection<ChartItem> Entries { get; set; }
```

Property Value

[ObservableCollection](#)<[ChartItem](#)>

## FillColor

Gets or sets the canvas default fill color. Default is transparent

```
public Color FillColor { get; set; }
```

Property Value

[Color](#)

## FontColor

Gets or sets the color to use when drawing value labels. Default is Black

```
public Color FontColor { get; set; }
```

Property Value

[Color](#)

## FontFamily

Gets or sets the canvas default Font to use when drawing strings

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

Property Value

[string](#)

## FontSize

Gets or sets font size to use when drawing value labels. Default is 11

```
public float FontSize { get; set; }
```

Property Value

[float](#)

## FooterLabelsFontColor

Gets or sets the color to use when drawing footer labels. Default is Black

```
public Color FooterLabelsFontColor { get; set; }
```

Property Value

[Color](#)

## FooterLabelsMargin

Gets or sets font margin to use when calculating footer labels coordinates. Default is 8

```
public float FooterLabelsMargin { get; set; }
```

Property Value

[float](#)

## FooterLabelsFontSize

Gets or sets font size to use for footer label values. Default is 10

```
public float FooterLabelsFontSize { get; set; }
```

Property Value

[float](#)

## HeaderValuesMargin

Gets or sets font margin to use when calculating header value labels coordinates. Default is 30

```
public float HeaderValuesMargin { get; set; }
```

Property Value

[float](#)

## LabelTextTruncationEnabled

If true, chart labels text will be truncated to fit available size. Default is true

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

Property Value

[bool](#)

## ItemSeparationMargin

Gets or sets the separation margin between each item. Default is 8

```
public float ItemSeparationMargin { get; set; }
```

Property Value

[float](#) ↗

## PathsColorOpacity

Sets the Alpha modifier value to use when drawing solid color backgrounds. Default is 0.6

```
public float PathsColorOpacity { get; set; }
```

Property Value

[float](#) ↗

## ReanimateOnPropertyChanged

If true, chart will trigger entry animation whenever a chart property is changed and EnableEntryAnimations prop is set to true. Default is true

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

Property Value

[bool](#) ↗

## StrokeColor

Stroke color. Default is Black

```
public Color StrokeColor { get; set; }
```

Property Value

[Color](#) ↗

## StrokeSize

Stroke size. Default is 2.5

```
public float StrokeSize { get; set; }
```

Property Value

[float](#)

## Methods

### Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public void Dispose()
```

# Class BaseChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public abstract class BaseChartDrawable : IDrawable, IDisposable
```

## Inheritance

[object](#) ← BaseChartDrawable

## Implements

[IDisposable](#), [IDisposable](#)

## Derived

[BarChartDrawable](#), [LineChartDrawable](#), [MultiBarChartDrawable](#), [MultiLineChartDrawable](#)

## Inherited Members

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

## Fields

### AxisXMargin

```
protected int AxisXMargin
```

#### Field Value

[int](#)

### IsAnimating

```
public bool IsAnimating
```

#### Field Value

[bool](#)

## IsInitialized

`public bool IsInitialized`

Field Value

[bool](#)

## MaxYValueCoordinate

`protected float MaxYValueCoordinate`

Field Value

[float](#)

## Points

`protected PointF[] Points`

Field Value

[PointF](#)[]

## \_colorBrush

`protected Brush _colorBrush`

Field Value

[Brush](#)

## \_dirtyRect

`protected RectF _dirtyRect`

### Field Value

[RectF](#)

## \_displayHeaderValues

`protected bool _displayHeaderValues`

### Field Value

[bool](#)

## \_displayValueLabelsOnTop

`protected bool _displayValueLabelsOnTop`

### Field Value

[bool](#)

## Properties

### AnimationProgress

Represents the percentage of the current chart Animation from 1-100

`public int AnimationProgress { get; set; }`

### Property Value

[int](#)

## AxisDashPattern

```
public float[] AxisDashPattern { get; set; }
```

Property Value

[float](#) []

## AxisFontSize

Gets or sets font size to use when drawing chart axis labels. Default is 11

```
public float AxisFontSize { get; set; }
```

Property Value

[float](#) []

## AxisLinesColor

Axis lines color. Default is Lightgray

```
public Color AxisLinesColor { get; set; }
```

Property Value

[Color](#) []

## AxisLinesStrokeSize

Axis lines stroke size. Default is 0.9

```
public float AxisLinesStrokeSize { get; set; }
```

Property Value

[float](#) ↗

## ColorBrush

`public Brush ColorBrush { get; set; }`

Property Value

[Brush](#) ↗

## DisplayHeaderValues

`public bool DisplayHeaderValues { get; set; }`

Property Value

[bool](#) ↗

## DisplayHorizontalAxisLines

If true header labels will be hidden and chart will draw horizontal step lines.

`public bool DisplayHorizontalAxisLines { get; set; }`

Property Value

[bool](#) ↗

## DisplayValueLabelsOnTop

If true, chart will show value labels on top of canvas. Default is true

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

Property Value

[bool](#) ↗

## DisplayVerticalAxisLines

If true vertical lines will be drawn as background along with horizontal lines. DisplayHorizontalAxisLines prop needs to be true as well

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

Property Value

[bool](#) ↗

## EnableAntialias

Gets or sets if the current canvas will use Antialias mode or not when drawing content. Default is true

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

Property Value

[bool](#) ↗

## Entries

Gets or sets the items to display within canvas

```
public ObservableCollection<ChartItem> Entries { get; set; }
```

Property Value

## [ObservableCollection](#) <ChartItem>

### FillColor

Gets or sets the canvas default fill color. Default is transparent

```
public Color FillColor { get; set; }
```

#### Property Value

[Color](#)

### Font

Gets or sets the canvas default Font to use when drawing strings. [10/10/22 -> GRAPHICS MISSING SUPPORT FOR CUSTOM FONTS]

```
public IFont Font { get; set; }
```

#### Property Value

[IFont](#)

### FontColor

Gets or sets the color to use when drawing value labels. Default is Black

```
public Color FontColor { get; set; }
```

#### Property Value

[Color](#)

### FontSize

Gets or sets font size to use when drawing value labels. Default is 11

```
public float FontSize { get; set; }
```

Property Value

[float](#)

## FooterHeight

Gets or sets Footer height

```
protected float FooterHeight { get; set; }
```

Property Value

[float](#)

## FooterLabelsFontColor

Gets or sets the color to use when drawing footer labels. Default is Black

```
public Color FooterLabelsFontColor { get; set; }
```

Property Value

[Color](#)

## FooterLabelsMargin

Gets or sets font margin to use when calculating footer labels coordinates. Default is 8

```
public float FooterLabelsMargin { get; set; }
```

Property Value

[float](#)

## FooterLabelsTextSize

Gets or sets font size to use for footer label values. Default is 10

```
public float FooterLabelsTextSize { get; set; }
```

Property Value

[float](#)

## HeaderHeight

Gets or sets Header height

```
protected float HeaderHeight { get; set; }
```

Property Value

[float](#)

## HeaderValuesMargin

Gets or sets font margin to use when calculating header value labels coordinates. Default is 30

```
public float HeaderValuesMargin { get; set; }
```

Property Value

[float](#)

## Height

Gets or sets control height

```
protected float Height { get; set; }
```

Property Value

[float](#)

## IsLabelTextTruncationEnabled

If true, chart labels text will be truncated to fit available size. Default is true

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

Property Value

[bool](#)

## ItemSeparationMargin

Gets or sets the separation margin between each item. Default is 8

```
public float ItemSeparationMargin { get; set; }
```

Property Value

[float](#)

## ItemSize

Gets or sets the size for each value column

```
protected SizeF ItemSize { get; set; }
```

Property Value

[SizeF](#)

## Margin

Gets or sets the default top and bottom internal margin to use when drawing canvas content. Default value is 15

```
public float Margin { get; set; }
```

### Property Value

[float](#)

## MaxValue

Gets the greatest value within ChartEntries collection

```
protected float MaxValue { get; }
```

### Property Value

[float](#)

## MinValue

Gets the smallest value within ChartEntries collection

```
protected float MinValue { get; }
```

### Property Value

[float](#)

## Origin

gets or sets the origin in Y axis

```
protected float Origin { get; set; }
```

Property Value

[float](#)

## PathsColorOpacity

Sets the Alpha modifier value to use when drawing solid color backgrounds.Default is 0.6

```
public float PathsColorOpacity { get; set; }
```

Property Value

[float](#)

## StrokeColor

Stroke color. Default is Black

```
public Color StrokeColor { get; set; }
```

Property Value

[Color](#)

## StrokeSize

Stroke size. Default is 2.5

```
public float StrokeSize { get; set; }
```

Property Value

[float](#)

## ValueRange

Gets or sets chart value range by subtracting MaxValue from MinValue.

```
protected float ValueRange { get; }
```

Property Value

[float](#)

## Width

Gets or sets control Width

```
protected float Width { get; set; }
```

Property Value

[float](#)

## \_canvas

```
protected ICanvas _canvas { get; set; }
```

Property Value

[ICanvas](#)

# Methods

## CalculateFooterHeight(SizeF[])

Calculates footer total height

```
protected virtual float CalculateFooterHeight(SizeF[] valueLabelSizes)
```

Parameters

**valueLabelSizes** [SizeF\[\]](#)[]

Label size

Returns

[float](#)

Labels height

## CalculateHeaderHeight(SizeF[])

Calculates header height

```
protected virtual float CalculateHeaderHeight(SizeF[] valueLabelSizes)
```

Parameters

**valueLabelSizes** [SizeF\[\]](#)[]

Label size

Returns

[float](#)

Header total height

## CalculateHorizontalYLineCoordinate(float, SizeF, float, float)

For a given value, calculates its Y coordinate within current canvas

```
protected virtual PointF CalculateHorizontalYLineCoordinate(float value, SizeF itemSize,  
float origin, float headerHeight)
```

Parameters

**value** [float](#)

Value to calculate

**itemSize** [SizeF](#)

Item available size

**origin** [float](#)

Y coordinate origin coordinate

**headerHeight** [float](#)

Header height

Returns

[PointF](#)

Y Coordinate of given value

## CalculateItemSize(List<ChartItem>, float, float, float, float)

Calculates the size chart is going to use when drawing each bar

```
protected virtual SizeF CalculateItemSize(List<ChartItem> entries, float width, float height, float footerHeight, float headerHeight)
```

Parameters

**entries** [List](#)<ChartItem>

**width** [float](#)

Control width

**height** [float](#)

Control height

**footerHeight** [float](#)

Chart footer label area height

**headerHeight** [float](#)

Chart header area height

Returns

### [SizeF](#)

The size for each bar

## CalculateItemSize(float, float, float, float)

Calculates the size chart is going to use when drawing each bar

```
protected virtual SizeF CalculateItemSize(float width, float height, float footerHeight,  
float headerHeight)
```

Parameters

### [width float](#)

Control width

### [height float](#)

Control height

### [footerHeight float](#)

Chart footer label area height

### [headerHeight float](#)

Chart header area height

Returns

### [SizeF](#)

The size for each bar

## CalculatePaddingForLabelValue(float, SizeF)

Calculate the center for a given rectangle

```
protected virtual float CalculatePaddingForLabelValue(float coordinateX, SizeF rect)
```

Parameters

**coordinateX** [float](#)

X axis coordinate

**rect** [SizeF](#)

Rectangle container to be centered

Returns

[float](#)

Number indicating the center X coordinate within input rectangle

## CalculatePoints(SizeF, float, float)

Calculates the coordinates of each value within the chart canvas

```
protected virtual PointF[] CalculatePoints(SizeF itemSize, float origin, float headerHeight)
```

Parameters

**itemSize** [SizeF](#)

Item size

**origin** [float](#)

Y axis origin coordinate

**headerHeight** [float](#)

Chart header area height

Returns

[PointF](#)[]

Array with the points coordinates

## CalculatePoints(List<ChartItem>, SizeF, float, float)

Calculates the coordinates of each value within the chart canvas

```
protected virtual PointF[] CalculatePoints(List<ChartItem> entries, SizeF itemSize, float origin, float headerHeight)
```

Parameters

`entries` [List](#)<ChartItem>

`itemSize` [SizeF](#)

Item size

`origin` [float](#)

Y axis origin coordinate

`headerHeight` [float](#)

Chart header area height

Returns

[PointF](#)[]

Array with the points coordinates

## CalculateYOrigin(float, float)

Calculates the Y axis origin value

```
protected float CalculateYOrigin(float itemHeight, float headerHeight)
```

Parameters

`itemHeight` [float](#)

Each bar width

**headerHeight** [float](#)

Control total height

Returns

[float](#)

Y axis origin coordinate

## ClearBackground(ICanvas, SizeF, Color)

Change current canvas background fill color

```
public virtual void ClearBackground(ICanvas canvas, SizeF containerSize, Color fillColor)
```

Parameters

**canvas** [ICanvas](#)

Current canvas

**containerSize** [SizeF](#)

Canvas size

**fillColor** [Color](#)

Fill color

## Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public virtual void Dispose()
```

## Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

## Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawChart(ICanvas, RectF)

Calculates and draws the current chart with the Entries[] data

```
public virtual void DrawChart(ICanvas canvas, RectF dirtyRect)
```

## Parameters

canvas [ICanvas](#)

Current canvas

dirtyRect [RectF](#)

Current canvas rect

## DrawHorizontalStepLines(ICanvas, IEnumerable<PointF>, float, float, float, float, float)

Draw horizontal lines in Y Axis

```
protected virtual void DrawHorizontalStepLines(ICanvas canvas, IEnumerable<PointF> points,  
float width, float height, float origin, float originX, float finalX)
```

## Parameters

canvas [ICanvas](#)

Current canvas

`points` [IEnumerable](#)<[PointF](#)>

Current chart calculated points

`width` [float](#)

Current canvas total Width

`height` [float](#)

Current Canvas total Height

`origin` [float](#)

Y origin coordinate

`originX` [float](#)

X coordinate that determines where the line should start

`finalX` [float](#)

X coordinate that determines where the line should end

## DrawLabels(ICanvas, PointF[], SizeF, float, float, float)

Draws footer and value labels on current canvas

```
protected virtual void DrawLabels(ICanvas canvas, PointF[] points, SizeF itemSize, float height, float footerHeight, float origin)
```

Parameters

`canvas` [ICanvas](#)

Canvas to draw on

`points` [PointF](#)[]

Coordinates for each footer label

`itemSize` [SizeF](#)

Item size

**height** [float](#)

Control height

**footerHeight** [float](#)

Chart footer area height

**origin** [float](#)

Y axis origin coordinate

**DrawLabels(List<ChartItem>, ICanvas, PointF[], SizeF, float, float, float)**

Draws footer and value labels on current canvas

```
protected virtual void DrawLabels(List<ChartItem> entries, ICanvas canvas, PointF[] points, SizeF itemSize, float height, float footerHeight, float origin)
```

Parameters

**entries** [List](#)<[ChartItem](#)>

**canvas** [ICanvas](#)

Canvas to draw on

**points** [PointF](#)[]

Coordinates for each footer label

**itemSize** [SizeF](#)

Item size

**height** [float](#)

Control height

**footerHeight** [float](#)

Chart footer area height

[origin](#) [float](#)

Y axis origin coordinate

## DrawVerticalStepLines(ICanvas, PointF[], float, float)

```
protected abstract void DrawVerticalStepLines(ICanvas canvas, PointF[] points, float origin,  
float maxPositiveValue)
```

### Parameters

[canvas](#) [ICanvas](#)

[points](#) [PointF](#)[]

[origin](#) [float](#)

[maxPositiveValue](#) [float](#)

## Entries\_CollectionChanged(object, NotifyCollectionChangedEventArgs)

```
protected virtual void Entries_CollectionChanged(object sender,  
NotifyCollectionChangedEventArgs e)
```

### Parameters

[sender](#) [object](#)

[e](#) [NotifyCollectionChangedEventArgs](#)

## Entry\_PropertyChanged(object, PropertyChangedEventArgs)

```
protected virtual void Entry_PropertyChanged(object sender, PropertyChangedEventArgs e)
```

### Parameters

sender [object](#)

e [PropertyChangedEventArgs](#)

## MeasureLabels()

Gets the size for labels container rectangle by measuring text length

```
protected virtual SizeF[] MeasureLabels()
```

Returns

[SizeF](#)[]

Array with labels rectangle containers

## RequestInvalidate()

Trigger Invalidate() method on ChartView()

```
protected void RequestInvalidate()
```

## TruncateText(ICanvas, SizeF, string)

Allows long words to be shortened by removing characters so it can fit within the provided area

```
protected string TruncateText(ICanvas canvas, SizeF availableSize, string text)
```

Parameters

canvas [ICanvas](#)

Canvas to draw on

availableSize [SizeF](#)

Represents the available width

**text** [string](#)

Text to truncate within the available area

Returns

[string](#)

Truncated text

# Class BusyIndicator

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The BusyIndicator is a drawn control that provides a graphical representation to indicate an application or process is busy.

```
public class BusyIndicator : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← BusyIndicator

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),

[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.Background](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.OnParentSet\(\)](#) , [NavigableElement.Navigation](#) , [NavigableElement.Style](#) ,  
[NavigableElement.StyleClass](#) , [NavigableElement.class](#) , [Element.AutomationIdProperty](#) ,  
[Element.ClassIdProperty](#) , [Element.ControlsElementMapper](#) ,  
[Element.InsertLogicalChild\(int, Element\)](#) , [Element.AddLogicalChild\(Element\)](#) ,  
[Element.RemoveLogicalChild\(Element\)](#) , [Element.ClearLogicalChildren\(\)](#) ,  
[Element.FindByName\(string\)](#) , [Element.RemoveDynamicResource\(BindableProperty\)](#) ,

[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### BusyIndicator()

```
public BusyIndicator()
```

## Fields

### BackgroundColorProperty

```
public static readonly BindableProperty BackgroundColorProperty
```

## Field Value

### [BindableProperty](#)

## ColorProperty

```
public static readonly BindableProperty ColorProperty
```

Field Value

[BindableProperty](#) ↗

## HasShadowProperty

```
public static readonly BindableProperty HasShadowProperty
```

Field Value

[BindableProperty](#) ↗

## ShadowColorProperty

```
public static readonly BindableProperty ShadowColorProperty
```

Field Value

[BindableProperty](#) ↗

## Properties

### BackgroundColor

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

Property Value

[Color](#)

## BusyIndicatorDrawable

```
public BusyIndicatorDrawable BusyIndicatorDrawable { get; set; }
```

Property Value

[BusyIndicatorDrawable](#)

## Color

```
public Color Color { get; set; }
```

Property Value

[Color](#)

## HasShadow

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

Property Value

[bool](#)

## ShadowColor

```
public Color ShadowColor { get; set; }
```

Property Value

[Color](#)

# Methods

## OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

### Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

# Class BusyIndicatorDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class BusyIndicatorDrawable : IDrawable
```

## Inheritance

[object](#) ← BusyIndicatorDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

## Constructors

### BusyIndicatorDrawable()

```
public BusyIndicatorDrawable()
```

## Properties

### BackgroundColor

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

Property Value

[Color](#)

### Color

```
public Color Color { get; set; }
```

Property Value

[Color](#) ↗

HasShadow

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

Property Value

[bool](#) ↗

Progress

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

Property Value

[double](#) ↗

Rotation

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

Property Value

[double](#) ↗

ShadowColor

```
public Color ShadowColor { get; set; }
```

## Property Value

[Color](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawArc(ICanvas, RectF)

```
public virtual void DrawArc(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawShadow(ICanvas, RectF)

```
public virtual void DrawShadow(ICanvas canvas, RectF dirtyRect)
```

### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

# Class Button

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Button represents a customizable drawn button control. It allows developers to create interactive buttons with highly customizable designs, leveraging the power of lightweight graphics rendering.

```
public class Button : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← [Button](#)

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),

[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BehaviorsProperty](#) , [VisualElement.TriggersProperty](#) ,  
[VisualElement.WidthRequestProperty](#) , [VisualElement.HeightRequestProperty](#) ,  
[VisualElement.MinimumWidthRequestProperty](#) , [VisualElement.MinimumHeightRequestProperty](#) ,  
[VisualElement.MaximumWidthRequestProperty](#) , [VisualElement.MaximumHeightRequestProperty](#) ,  
[VisualElement.IsFocusedProperty](#) , [VisualElement.FlowDirectionProperty](#) ,  
[VisualElement.WindowProperty](#) , [VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.Behaviors](#) , [VisualElement.Bounds](#) , [VisualElement.Height](#) ,  
[VisualElement.HeightRequest](#) , [VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) ,  
[VisualElement.IsEnabledCore](#) , [VisualElement.IsFocused](#) , [VisualElement.Visibile](#) ,  
[VisualElement.MinimumHeightRequest](#) , [VisualElement.MinimumWidthRequest](#) ,  
[VisualElement.MaximumHeightRequest](#) , [VisualElement.MaximumWidthRequest](#) ,  
[VisualElement.Opacity](#) , [VisualElement.Rotation](#) , [VisualElement.RotationX](#) ,  
[VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) , [VisualElement.ScaleY](#) ,  
[VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) , [VisualElement.Triggers](#) ,  
[VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) , [VisualElement.Y](#) ,  
[VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) , [VisualElement.Handler](#) ,  
[VisualElement.Shadow](#) , [VisualElement.ZIndex](#) , [VisualElement.DesiredSize](#) ,  
[VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) , [VisualElement.Focused](#) ,  
[VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) , [VisualElement.Unfocused](#) ,  
[VisualElement.Loaded](#) , [VisualElement.Unloaded](#) , [NavigableElement.OnParentSet\(\)](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,

[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### Button()

```
public Button()
```

## Fields

### BackgroundColorProperty

```
public static readonly BindableProperty BackgroundColorProperty
```

## Field Value

[BindableProperty](#)

## BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

Field Value

[BindableProperty ↗](#)

## CommandParameterProperty

```
public static readonly BindableProperty CommandParameterProperty
```

Field Value

[BindableProperty ↗](#)

## CommandProperty

```
public static readonly BindableProperty CommandProperty
```

Field Value

[BindableProperty ↗](#)

## CornerRadiusProperty

```
public static readonly BindableProperty CornerRadiusProperty
```

Field Value

[BindableProperty ↗](#)

## FontSizeProperty

```
public static readonly BindableProperty FontSizeProperty
```

Field Value

[BindableProperty](#)

## HasShadowProperty

```
public static readonly BindableProperty HasShadowProperty
```

Field Value

[BindableProperty](#)

## HorizontalTextAlignmentProperty

```
public static readonly BindableProperty HorizontalTextAlignmentProperty
```

Field Value

[BindableProperty](#)

## PressedCommandParameterProperty

```
public static readonly BindableProperty PressedCommandParameterProperty
```

Field Value

[BindableProperty](#)

## PressedCommandProperty

```
public static readonly BindableProperty PressedCommandProperty
```

Field Value

[BindableProperty](#)

## ReleasedCommandParameterProperty

```
public static readonly BindableProperty ReleasedCommandParameterProperty
```

Field Value

[BindableProperty](#)

## ReleasedCommandProperty

```
public static readonly BindableProperty ReleasedCommandProperty
```

Field Value

[BindableProperty](#)

## ShadowColorProperty

```
public static readonly BindableProperty ShadowColorProperty
```

Field Value

[BindableProperty](#)

## StrokeProperty

```
public static readonly BindableProperty StrokeProperty
```

Field Value

[BindableProperty](#)

## StrokeThicknessProperty

```
public static readonly BindableProperty StrokeThicknessProperty
```

Field Value

[BindableProperty](#)

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty](#)

## TextProperty

```
public static readonly BindableProperty TextProperty
```

Field Value

[BindableProperty](#)

## VerticalTextAlignmentProperty

```
public static readonly BindableProperty VerticalTextAlignmentProperty
```

Field Value

[BindableProperty](#)

## Properties

Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

BackgroundColor

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

Property Value

[Color](#)

ButtonDrawable

```
public ButtonDrawable ButtonDrawable { get; set; }
```

Property Value

[ButtonDrawable](#)

Command

```
public ICommand Command { get; set; }
```

Property Value

[ICommand](#) ↗

## CommandParameter

```
public object CommandParameter { get; set; }
```

Property Value

[object](#) ↗

## CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

Property Value

[CornerRadius](#) ↗

## FontSize

```
public float FontSize { get; set; }
```

Property Value

[float](#) ↗

## HasShadow

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

Property Value

[bool](#)

## HorizontalTextAlignment

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

Property Value

[TextAlignment](#)

## PressedCommand

```
public ICommand PressedCommand { get; set; }
```

Property Value

[ICommand](#)

## PressedCommandParameter

```
public object PressedCommandParameter { get; set; }
```

Property Value

[object](#)

## ReleasedCommand

```
public ICommand ReleasedCommand { get; set; }
```

Property Value

[ICommand](#)

## ReleasedCommandParameter

```
public object ReleasedCommandParameter { get; set; }
```

Property Value

[object](#)

## ShadowColor

```
public Color ShadowColor { get; set; }
```

Property Value

[Color](#)

## Stroke

```
public Brush Stroke { get; set; }
```

Property Value

[Brush](#)

## StrokeThickness

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

Property Value

[double](#)

## Text

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

Property Value

[string](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#)

## VerticalTextAlignment

```
public TextAlignement VerticalTextAlignment { get; set; }
```

Property Value

[TextAlignement](#)

## Methods

### OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

## Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

# Events

## Clicked

```
public event EventHandler Clicked
```

### Event Type

[EventHandler](#)

## Pressed

```
public event EventHandler Pressed
```

### Event Type

[EventHandler](#)

## Released

```
public event EventHandler Released
```

### Event Type

[EventHandler](#)

# Class ButtonDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class ButtonDrawable : IDrawable
```

## Inheritance

[object](#) ← ButtonDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

## Constructors

### ButtonDrawable()

```
public ButtonDrawable()
```

## Properties

### BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

Property Value

[Paint](#)

### CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

Property Value

[CornerRadius](#)

## FontSize

```
public float FontSize { get; set; }
```

Property Value

[float](#)

## HasShadow

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

Property Value

[bool](#)

## HorizontalTextAlignment

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

Property Value

[TextAlignment](#)

## ShadowColor

```
public Color ShadowColor { get; set; }
```

Property Value

[Color](#)

## StrokePaint

```
public Paint StrokePaint { get; set; }
```

Property Value

[Paint](#)

## StrokeThickness

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

Property Value

[double](#)

## Text

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

Property Value

[string](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#)

## VerticalTextAlignment

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

Property Value

[TextAlignment](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawRippleEffect(ICanvas, RectF)

```
public virtual void DrawRippleEffect(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawShadow(ICanvas, RectF)

```
public virtual void DrawShadow(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawStroke(ICanvas, RectF)

```
public virtual void DrawStroke(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawText(ICanvas, RectF)

```
public virtual void DrawText(ICanvas canvas, RectF dirtyRect)
```

## Parameters

**canvas** [ICanvas](#) ↗

**dirtyRect** [RectF](#) ↗

# Class Captcha

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Captcha is a drawn control used to render and manage CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenges. It leverages .NET MAUI Graphics for efficient graphical rendering and provides a flexible way to generate and validate CAPTCHAs in an application.

```
public class Captcha : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← Captcha
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
```

[VisualElement.RotationXProperty](#) , [VisualElement.RotationYProperty](#) , [VisualElement.ScaleProperty](#) ,  
[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BackgroundProperty](#) ,  
[VisualElement.BehaviorsProperty](#) , [VisualElement.TriggersProperty](#) ,  
[VisualElement.WidthRequestProperty](#) , [VisualElement.HeightRequestProperty](#) ,  
[VisualElement.MinimumWidthRequestProperty](#) , [VisualElement.MinimumHeightRequestProperty](#) ,  
[VisualElement.MaximumWidthRequestProperty](#) , [VisualElement.MaximumHeightRequestProperty](#) ,  
[VisualElement.IsFocusedProperty](#) , [VisualElement.FlowDirectionProperty](#) ,  
[VisualElement.WindowProperty](#) , [VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Background](#) ,  
[VisualElement.Behaviors](#) , [VisualElement.Bounds](#) , [VisualElement.Height](#) ,  
[VisualElement.HeightRequest](#) , [VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) ,  
[VisualElement.IsEnabledCore](#) , [VisualElement.IsFocused](#) , [VisualElement.Visibile](#) ,  
[VisualElement.MinimumHeightRequest](#) , [VisualElement.MinimumWidthRequest](#) ,  
[VisualElement.MaximumHeightRequest](#) , [VisualElement.MaximumWidthRequest](#) ,  
[VisualElement.Opacity](#) , [VisualElement.Rotation](#) , [VisualElement.RotationX](#) ,  
[VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) , [VisualElement.ScaleY](#) ,  
[VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) , [VisualElement.Triggers](#) ,  
[VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) , [VisualElement.Y](#) ,  
[VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) , [VisualElement.Handler](#) ,  
[VisualElement.Shadow](#) , [VisualElement.ZIndex](#) , [VisualElement.DesiredSize](#) ,  
[VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) , [VisualElement.Focused](#) ,  
[VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) , [VisualElement.Unfocused](#) ,  
[VisualElement.Loaded](#) , [VisualElement.Unloaded](#) , [NavigableElement.OnParentSet\(\)](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,

[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### Captcha()

```
public Captcha()
```

## Fields

### LevelProperty

```
public static readonly BindableProperty LevelProperty
```

Field Value

[BindableProperty](#) ↗

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty](#) ↗

## Properties

### CaptchaDrawable

```
public CaptchaDrawable CaptchaDrawable { get; set; }
```

Property Value

[CaptchaDrawable](#)

### Level

```
public CaptchaLevel Level { get; set; }
```

Property Value

[CaptchaLevel](#)

### TextColor

```
public Color TextColor { get; set; }
```

## Property Value

[Color](#) ↗

## Methods

### OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) ↗ event.

```
protected override void OnParentChanged()
```

### Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) ↗ event.

# Class CaptchaDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class CaptchaDrawable : IDrawable
```

## Inheritance

[object](#) ← CaptchaDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## Level

```
public CaptchaLevel Level { get; set; }
```

## Property Value

[CaptchaLevel](#)

## TextColor

```
public Color TextColor { get; set; }
```

## Property Value

[Color](#)

# Word

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

## Property Value

[string](#)

# Methods

## Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawArtifacts(ICanvas, RectF)

```
public virtual void DrawArtifacts(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawText(ICanvas, RectF)

```
public virtual void DrawText(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Enum CaptchaLevel

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The CaptchaLevel enumeration defines the different levels of complexity for CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenges. This can be used to configure or determine the strength of a CAPTCHA system, balancing between ease of use and security requirements.

```
public enum CaptchaLevel
```

## Fields

Normal = 1

Strong = 2

Weak = 0

# Class CheckBox

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The CheckBox is a custom drawn control that represents a checkbox for selecting or deselecting options. It is highly customizable while maintaining efficient rendering capabilities.

```
public class CheckBox : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← [CheckBox](#)

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),

[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BackgroundProperty](#) ,  
[VisualElement.BehaviorsProperty](#) , [VisualElement.TriggersProperty](#) ,  
[VisualElement.WidthRequestProperty](#) , [VisualElement.HeightRequestProperty](#) ,  
[VisualElement.MinimumWidthRequestProperty](#) , [VisualElement.MinimumHeightRequestProperty](#) ,  
[VisualElement.MaximumWidthRequestProperty](#) , [VisualElement.MaximumHeightRequestProperty](#) ,  
[VisualElement.IsFocusedProperty](#) , [VisualElement.FlowDirectionProperty](#) ,  
[VisualElement.WindowProperty](#) , [VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Background](#) ,  
[VisualElement.Behaviors](#) , [VisualElement.Bounds](#) , [VisualElement.Height](#) ,  
[VisualElement.HeightRequest](#) , [VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) ,  
[VisualElement.IsEnabledCore](#) , [VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) ,  
[VisualElement.MinimumHeightRequest](#) , [VisualElement.MinimumWidthRequest](#) ,  
[VisualElement.MaximumHeightRequest](#) , [VisualElement.MaximumWidthRequest](#) ,  
[VisualElement.Opacity](#) , [VisualElement.Rotation](#) , [VisualElement.RotationX](#) ,  
[VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) , [VisualElement.ScaleY](#) ,  
[VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) , [VisualElement.Triggers](#) ,  
[VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) , [VisualElement.Y](#) ,  
[VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) , [VisualElement.Handler](#) ,  
[VisualElement.Shadow](#) , [VisualElement.ZIndex](#) , [VisualElement.DesiredSize](#) ,  
[VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) , [VisualElement.Focused](#) ,  
[VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) , [VisualElement.Unfocused](#) ,  
[VisualElement.Loaded](#) , [VisualElement.Unloaded](#) , [NavigableElement.OnParentSet\(\)](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### CheckBox()

```
public CheckBox()
```

## Fields

### CheckedBrushProperty

```
public static readonly BindableProperty CheckedBrushProperty
```

## Field Value

[BindableProperty](#)

## CheckedCommandParameterProperty

```
public static readonly BindableProperty CheckedCommandParameterProperty
```

Field Value

[BindableProperty](#)

## CheckedCommandProperty

```
public static readonly BindableProperty CheckedCommandProperty
```

Field Value

[BindableProperty](#)

## IsCheckedProperty

```
public static readonly BindableProperty IsCheckedProperty
```

Field Value

[BindableProperty](#)

## StrokeProperty

```
public static readonly BindableProperty StrokeProperty
```

Field Value

[BindableProperty](#)

## StrokeThicknessProperty

```
public static readonly BindableProperty StrokeThicknessProperty
```

Field Value

[BindableProperty↗](#)

## UncheckedBrushProperty

```
public static readonly BindableProperty UncheckedBrushProperty
```

Field Value

[BindableProperty↗](#)

## UncheckedCommandParameterProperty

```
public static readonly BindableProperty UncheckedCommandParameterProperty
```

Field Value

[BindableProperty↗](#)

## UncheckedCommandProperty

```
public static readonly BindableProperty UncheckedCommandProperty
```

Field Value

[BindableProperty↗](#)

## Properties

## CheckBoxDrawable

```
public CheckBoxDrawable CheckBoxDrawable { get; set; }
```

Property Value

[CheckBoxDrawable](#)

## CheckedBrush

```
public Brush CheckedBrush { get; set; }
```

Property Value

[Brush](#)

## CheckedCommand

```
public ICommand CheckedCommand { get; set; }
```

Property Value

[ICommand](#)

## CheckedCommandParameter

```
public object CheckedCommandParameter { get; set; }
```

Property Value

[object](#)

## IsChecked

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

Property Value

[bool](#)

Stroke

```
public Brush Stroke { get; set; }
```

Property Value

[Brush](#)

StrokeThickness

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

Property Value

[double](#)

UncheckedBrush

```
public Brush UncheckedBrush { get; set; }
```

Property Value

[Brush](#)

UncheckedCommand

```
public ICommand UncheckedCommand { get; set; }
```

Property Value

[ICommand](#)

## UncheckedCommandParameter

```
public object UncheckedCommandParameter { get; set; }
```

Property Value

[object](#)

## Methods

### OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

## Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

## Events

### CheckedChanged

```
public event EventHandler<CheckedChangedEventArgs> CheckedChanged
```

Event Type

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



# Class CheckBoxDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class CheckBoxDrawable : IDrawable
```

## Inheritance

[object](#) ← CheckBoxDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## CheckedPaint

```
public Paint CheckedPaint { get; set; }
```

### Property Value

[Paint](#)

## IsChecked

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

### Property Value

[bool](#)

## StrokePaint

```
public Paint StrokePaint { get; set; }
```

Property Value

[Paint](#)

## StrokeThickness

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

Property Value

[double](#)

## UncheckedPaint

```
public Paint UncheckedPaint { get; set; }
```

Property Value

[Paint](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawCheck(ICanvas, RectF)

```
public virtual void DrawCheck(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Class LineChart

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The LineChart provides a drawn control to visualize data as a line chart. It allows data points to be represented by a continuous line, making it ideal for showing trends over time or relationships between variables.

This class inherits from BaseChart, leveraging shared charting functionality while adding line-specific rendering features.

```
public sealed class LineChart : BaseChart, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform, IDisposable
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← [BaseChart](#) ← LineChart

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#),  
[IDisposable](#)

## Inherited Members

[BaseChart.EasingProperty](#), [BaseChart.Easing](#), [BaseChart.BackgroundProperty](#), [BaseChart.Background](#),  
[BaseChart.AxisDashPatternProperty](#), [BaseChart.AxisDashPattern](#),  
[BaseChart.DisplayHeaderValuesProperty](#), [BaseChart.DisplayHeaderValues](#),  
[BaseChart.PathsColorOpacityProperty](#), [BaseChart.PathsColorOpacity](#),  
[BaseChart.AxisLinesFontSizeProperty](#), [BaseChart.AxisLinesFontSize](#), [BaseChart.AxisLineColorProperty](#),  
[BaseChart.AxisLinesColor](#), [BaseChart.AxisLinesStrokeSizeProperty](#), [BaseChart.AxisLinesStrokeSize](#),  
[BaseChart.DisplayVerticalAxisLinesProperty](#), [BaseChart.DisplayVerticalAxisLines](#),  
[BaseChart.DisplayHorizontalAxisLinesProperty](#), [BaseChart.DisplayHorizontalAxisLines](#),  
[BaseChart.AnimationIntervalProperty](#), [BaseChart.AnimationInterval](#),  
[BaseChart.ReanimateOnPropertyChangedProperty](#), [BaseChart.ReanimateOnPropertyChanged](#),

[BaseChart.EnableEntryAnimationsProperty](#), [BaseChart.EnableEntryAnimations](#),  
[BaseChart.EntriesProperty](#), [BaseChart.Entries](#), [BaseChart.IsLabelTextTruncationEnabledProperty](#),  
[BaseChart.IsLabelTextTruncationEnabled](#), [BaseChart.DisplayValueLabelsOnTopProperty](#),  
[BaseChart.DisplayValueLabelsOnTop](#), [BaseChart.StrokeSizeProperty](#), [BaseChart.StrokeSize](#),  
[BaseChart.StrokeColorProperty](#), [BaseChart.StrokeColor](#), [BaseChart.ItemSeparationMarginProperty](#),  
[BaseChart.ItemSeparationMargin](#), [BaseChart.FooterLabelsTextSizeProperty](#),  
[BaseChart.FooterLabelsTextSize](#), [BaseChart.FooterLabelsMarginProperty](#),  
[BaseChart.FooterLabelsMargin](#), [BaseChart.HeaderValuesMarginProperty](#),  
[BaseChart.HeaderValuesMargin](#), [BaseChart.FillColorProperty](#), [BaseChart.FillColor](#),  
[BaseChart.FontFamilyProperty](#), [BaseChart.FontFamily](#), [BaseChart.FontSizeProperty](#), [BaseChart.FontSize](#),  
[BaseChart.ChartMarginProperty](#), [BaseChart.ChartMargin](#), [BaseChart.FontColorProperty](#),  
[BaseChart.FontColor](#), [BaseChart.FooterLabelsFontColorProperty](#), [BaseChart.FooterLabelsFontColor](#),  
[BaseChart.EnableAntialiasProperty](#), [BaseChart.EnableAntialias](#), [BaseChart.Dispose\(\)](#),  
[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.GetChildElements\(Point\)](#),  
[View.GestureRecognizers](#), [View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#),  
[VisualElement.NavigationProperty](#), [VisualElement.StyleProperty](#),  
[VisualElement.InputTransparentProperty](#), [VisualElement.IsEnabledProperty](#),  
[VisualElement.XProperty](#), [VisualElement.YProperty](#), [VisualElement.AnchorXProperty](#),  
[VisualElement.AnchorYProperty](#), [VisualElement.TranslationXProperty](#),  
[VisualElement.TranslationYProperty](#), [VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#),  
[VisualElement.RotationProperty](#), [VisualElement.RotationXProperty](#),  
[VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#), [VisualElement.ScaleXProperty](#),  
[VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#), [VisualElement.VisualProperty](#),  
[VisualElement.IsCheckedProperty](#), [VisualElement.OpacityProperty](#),  
[VisualElement.BackgroundColorProperty](#), [VisualElement.BehaviorsProperty](#),  
[VisualElement.TriggersProperty](#), [VisualElement.WidthRequestProperty](#),  
[VisualElement.HeightRequestProperty](#), [VisualElement.MinimumWidthRequestProperty](#),  
[VisualElement.MinimumHeightRequestProperty](#), [VisualElement.MaximumWidthRequestProperty](#),  
[VisualElement.MaximumHeightRequestProperty](#), [VisualElement.IsFocusedProperty](#),  
[VisualElement.FlowDirectionProperty](#), [VisualElement.WindowProperty](#),  
[VisualElement.ShadowProperty](#), [VisualElement.ZIndexProperty](#),  
[VisualElement.ControlsVisualElementMapper](#), [VisualElement.BatchBegin\(\)](#),  
[VisualElement.BatchCommit\(\)](#), [VisualElement.Focus\(\)](#),  
[VisualElement.Measure\(double, double, MeasureFlags\)](#), [VisualElement.Unfocus\(\)](#),  
[VisualElement.Arrange\(Rect\)](#), [VisualElement.Layout\(Rect\)](#),  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#),

[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsFocused](#) ,  
[VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## LineChart()

```
public LineChart()
```

# Fields

## ChartStyleProperty

```
public static readonly BindableProperty ChartStyleProperty
```

Field Value

[BindableProperty](#)

## CurveFactorProperty

```
public static readonly BindableProperty CurveFactorProperty
```

Field Value

[BindableProperty](#)

## ExpandAndFillBackgroundCurvePathProperty

```
public static readonly BindableProperty ExpandAndFillBackgroundCurvePathProperty
```

Field Value

[BindableProperty](#)

## FillCurveColorProperty

```
public static readonly BindableProperty FillCurveColorProperty
```

Field Value

[BindableProperty](#)

## IsCurveBackgroundFilledProperty

```
public static readonly BindableProperty IsCurveBackgroundFilledProperty
```

Field Value

[BindableProperty](#)

## LineColorProperty

```
public static readonly BindableProperty LineColorProperty
```

Field Value

[BindableProperty](#)

## PointColorProperty

```
public static readonly BindableProperty PointColorProperty
```

Field Value

[BindableProperty](#)

## PointSizeProperty

```
public static readonly BindableProperty PointSizeProperty
```

Field Value

[BindableProperty](#)

## ShowPointsForCurveStyleProperty

```
public static readonly BindableProperty ShowPointsForCurveStyleProperty
```

Field Value

[BindableProperty](#)

## Properties

### ChartStyle

Defines the style for the current LineChart

```
public ChartEnums.LineChartStyle ChartStyle { get; set; }
```

Property Value

[ChartEnums.LineChartStyle](#)

### CurveFactor

Sets how 'curvy' the bezier curve will be when drawn. Accepts values between 0-1. Default is 0.6

```
public float CurveFactor { get; set; }
```

Property Value

[float](#)

## ExpandAndFillBackgroundCurvePath

If true Bezier curve Background Path color will expand and fill when drawn.

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

Property Value

[bool](#)

## FillCurveColor

Gets or sets the color to use when filling Curve serie

```
public Color FillCurveColor { get; set; }
```

Property Value

[Color](#)

## IsCurveBackgroundFilled

Gets or sets if the curve path will have a solid color background when drawn.

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

Property Value

[bool](#)

## LineColor

Gets or sets the color to use when drawing chart lines.

```
public Color LineColor { get; set; }
```

Property Value

[Color](#) ↗

## PointColor

Gets or sets the color for each dot.

```
public Color PointColor { get; set; }
```

Property Value

[Color](#) ↗

## PointSize

Gets or sets the radius for each point.

```
public float PointSize { get; set; }
```

Property Value

[float](#) ↗

## ShowPointsForCurveStyle

If true and Style prop equals to 'Curve' points will be shown when drawing cubic bezier

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

Property Value

[bool](#) ↗

# Class LineChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public sealed class LineChartDrawable : BaseChartDrawable, IDrawable, IDisposable
```

## Inheritance

[object](#) ← [BaseChartDrawable](#) ← LineChartDrawable

## Implements

[IDisposable](#), [IDisposable](#)

## Inherited Members

[BaseChartDrawable.Initialized](#), [BaseChartDrawable.IsAnimating](#), [BaseChartDrawable.ColorBrush](#),  
[BaseChartDrawable.AxisDashPattern](#), [BaseChartDrawable.DisplayHeaderValues](#),  
[BaseChartDrawable.PathsColorOpacity](#), [BaseChartDrawable.DisplayHorizontalAxisLines](#),  
[BaseChartDrawable.DisplayVerticalAxisLines](#), [BaseChartDrawable.AnimationProgress](#),  
[BaseChartDrawable.LabelTextTruncationEnabled](#), [BaseChartDrawable.DisplayValueLabelsOnTop](#),  
[BaseChartDrawable.StrokeSize](#), [BaseChartDrawable.AxisLinesStrokeSize](#),  
[BaseChartDrawable.AxisLinesColor](#), [BaseChartDrawable.StrokeColor](#),  
[BaseChartDrawable.ItemSeparationMargin](#), [BaseChartDrawable.FooterLabelsTextSize](#),  
[BaseChartDrawable.FooterLabelsMargin](#), [BaseChartDrawable.HeaderValuesMargin](#),  
[BaseChartDrawable.FillColor](#), [BaseChartDrawable.Font](#), [BaseChartDrawable.Entries](#),  
[BaseChartDrawable.FontSize](#), [BaseChartDrawable.AxisFontSize](#), [BaseChartDrawable.Margin](#),  
[BaseChartDrawable.FontColor](#), [BaseChartDrawable.FooterLabelsFontColor](#),  
[BaseChartDrawable.EnableAntialias](#), [BaseChartDrawable.Draw\(ICanvas, RectF\)](#),  
[BaseChartDrawable.ClearBackground\(ICanvas, SizeF, Color\)](#), [BaseChartDrawable.Dispose\(\)](#),  
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

# Properties

## CurveFactor

Sets how 'curvy' the bezier curve will be when drawn. Accepts values between 0-1. Default is 0.6

```
public float CurveFactor { get; set; }
```

Property Value

[float](#)

## ExpandAndFillBackgroundCurvePath

If true Bezier curve Background Path color will expand and fill when drawn.

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

Property Value

[bool](#)

## FillCurveColor

Gets or sets the color to use when filling Curve serie

```
public Color FillCurveColor { get; set; }
```

Property Value

[Color](#)

## IsCurveBackgroundFilled

Gets or sets if the curve will have a solid color background when drawn.

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

Property Value

[bool](#)

## LineColor

Gets or sets the color to use when drawing chart lines.

```
public Color LineColor { get; set; }
```

Property Value

[Color](#) ↗

## PointColor

Gets or sets the color for each dot.

```
public Color PointColor { get; set; }
```

Property Value

[Color](#) ↗

## PointSize

Gets or sets the radius for each point.

```
public float PointSize { get; set; }
```

Property Value

[float](#) ↗

## ShowPointsForCurveStyle

If true and Style prop equals to 'Curve' points will be shown when drawing cubic bezier

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

Property Value

[bool](#)

## Style

Defines the style for the current LineChart

```
public ChartEnums.LineChartStyle Style { get; set; }
```

Property Value

[ChartEnums.LineChartStyle](#)

## Methods

### DrawChart(ICanvas, RectF)

Calculates and draws the current chart with the Entries[] data

```
public override void DrawChart(ICanvas canvas, RectF dirtyRect)
```

Parameters

[canvas](#) [ICanvas](#)

Current canvas

[dirtyRect](#) [RectF](#)

Current canvas rect

### DrawVerticalStepLines(ICanvas, PointF[], float, float)

```
protected override void DrawVerticalStepLines(ICanvas canvas, PointF[] points, float origin, float maxPositiveValue)
```

Parameters

canvas [ICanvas](#)

points [PointF](#)[]

origin [float](#)

maxPositiveValue [float](#)

# Class LinearGauge

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The LinearGauge is a drawn control for rendering a linear gauge. A linear gauge is a visual representation of data along a straight or horizontal/vertical axis, commonly used to display values such as progress, measurements, or performance indicators.

```
public class LinearGauge : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← LinearGauge
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
```

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### LinearGauge()

```
public LinearGauge()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

[BindableProperty](#)

## CornerRadiusProperty

```
public static readonly BindableProperty CornerRadiusProperty
```

Field Value

[BindableProperty](#)

## RangeEndProperty

```
public static readonly BindableProperty RangeEndProperty
```

Field Value

[BindableProperty](#)

## RangeStartProperty

```
public static readonly BindableProperty RangeStartProperty
```

Field Value

[BindableProperty](#)

## ValueProperty

```
public static readonly BindableProperty ValueProperty
```

Field Value

[BindableProperty](#)

# Properties

## Background

```
public Brush Background { get; set; }
```

### Property Value

[Brush](#) ↗

## CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

### Property Value

[CornerRadius](#) ↗

## LinearGaugeDrawable

```
public LinearGaugeDrawable LinearGaugeDrawable { get; set; }
```

### Property Value

[LinearGaugeDrawable](#)

## RangeEnd

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

### Property Value

[int](#) ↗

## RangeStart

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

Property Value

[int↗](#)

## Value

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

Property Value

[int↗](#)

## Methods

### OnParentSet()

Raises the (internal) [ParentSet](#) event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet↗\(\)](#)

## Events

### ValueChanged

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

## Event Type

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

# Class LinearGaugeDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class LinearGaugeDrawable : IDrawable
```

## Inheritance

[object](#) ← LinearGaugeDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

### Property Value

[CornerRadius](#)

## RangeEnd

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

Property Value

[int↗](#)

## RangeStart

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

Property Value

[int↗](#)

## Value

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

Property Value

[int↗](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas↗](#)

dirtyRect [RectF↗](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawProgress(ICanvas, RectF)

```
public virtual void DrawProgress(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawTicks(ICanvas, RectF)

```
public virtual void DrawTicks(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Class MultiBarChart

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The MultiBarChart is a drawn control designed to display multiple bar charts within a single charting view. It enables the visualization of grouped or stacked data, making it an ideal tool for comparing multiple datasets across categories.

This class inherits from BaseChart, leveraging core charting functionality while adding features specific to bar chart rendering.

```
public sealed class MultiBarChart : BaseChart, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform, IDisposable
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← [BaseChart](#) ← MultiBarChart

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#),  
[IDisposable](#)

## Inherited Members

[BaseChart.EasingProperty](#), [BaseChart.Easing](#), [BaseChart.BackgroundProperty](#), [BaseChart.Background](#),  
[BaseChart.AxisDashPatternProperty](#), [BaseChart.AxisDashPattern](#),  
[BaseChart.DisplayHeaderValuesProperty](#), [BaseChart.DisplayHeaderValues](#),  
[BaseChart.PathsColorOpacityProperty](#), [BaseChart.PathsColorOpacity](#),  
[BaseChart.AxisLinesFontSizeProperty](#), [BaseChart.AxisLinesFontSize](#), [BaseChart.AxisLineColorProperty](#),  
[BaseChart.AxisLinesColor](#), [BaseChart.AxisLinesStrokeSizeProperty](#), [BaseChart.AxisLinesStrokeSize](#),  
[BaseChart.DisplayVerticalAxisLinesProperty](#), [BaseChart.DisplayVerticalAxisLines](#),  
[BaseChart.DisplayHorizontalAxisLinesProperty](#), [BaseChart.DisplayHorizontalAxisLines](#),  
[BaseChart.AnimationIntervalProperty](#), [BaseChart.AnimationInterval](#),  
[BaseChart.ReanimateOnPropertyChangedProperty](#), [BaseChart.ReanimateOnPropertyChanged](#),

[BaseChart.EnableEntryAnimationsProperty](#), [BaseChart.EnableEntryAnimations](#),  
[BaseChart.IsLabelTextTruncationEnabledProperty](#), [BaseChart.IsLabelTextTruncationEnabled](#),  
[BaseChart.DisplayValueLabelsOnTopProperty](#), [BaseChart.DisplayValueLabelsOnTop](#),  
[BaseChart.StrokeSizeProperty](#), [BaseChart.StrokeSize](#), [BaseChart.StrokeColorProperty](#),  
[BaseChart.StrokeColor](#), [BaseChart.FooterLabelsTextSizeProperty](#), [BaseChart.FooterLabelsTextSize](#),  
[BaseChart.FooterLabelsMarginProperty](#), [BaseChart.FooterLabelsMargin](#),  
[BaseChart.HeaderValuesMarginProperty](#), [BaseChart.HeaderValuesMargin](#), [BaseChart.FillColorProperty](#),  
[BaseChart.FillColor](#), [BaseChart.FontFamilyProperty](#), [BaseChart.FontFamily](#), [BaseChart.FontSizeProperty](#),  
[BaseChart.FontSize](#), [BaseChart.ChartMarginProperty](#), [BaseChart.ChartMargin](#),  
[BaseChart.FontColorProperty](#), [BaseChart.FontColor](#), [BaseChart.FooterLabelsFontColorProperty](#),  
[BaseChart.FooterLabelsFontColor](#), [BaseChart.EnableAntialiasProperty](#), [BaseChart.EnableAntialias](#),  
[BaseChart.Dispose\(\)](#), [GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#),  
[GraphicsView.Drawable](#), [GraphicsView.StartHoverInteraction](#),  
[GraphicsView.MoveHoverInteraction](#), [GraphicsView.EndHoverInteraction](#),  
[GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#), [GraphicsView.EndInteraction](#),  
[GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#), [View.HorizontalOptionsProperty](#),  
[View.MarginProperty](#), [View.GetChildElements\(Point\)](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),  
[VisualElement.VisualProperty](#), [VisualElement.IsVisibleProperty](#), [VisualElement.OpacityProperty](#),  
[VisualElement.BackgroundColorProperty](#), [VisualElement.BehaviorsProperty](#),  
[VisualElement.TriggersProperty](#), [VisualElement.WidthRequestProperty](#),  
[VisualElement.HeightRequestProperty](#), [VisualElement.MinimumWidthRequestProperty](#),  
[VisualElement.MinimumHeightRequestProperty](#), [VisualElement.MaximumWidthRequestProperty](#),  
[VisualElement.MaximumHeightRequestProperty](#), [VisualElement.FocusedProperty](#),  
[VisualElement.FlowDirectionProperty](#), [VisualElement.WindowProperty](#),  
[VisualElement.ShadowProperty](#), [VisualElement.ZIndexProperty](#),  
[VisualElement.ControlsVisualElementMapper](#), [VisualElement.BatchBegin\(\)](#),  
[VisualElement.BatchCommit\(\)](#), [VisualElement.Focus\(\)](#),  
[VisualElement.Measure\(double, double, MeasureFlags\)](#), [VisualElement.Unfocus\(\)](#),  
[VisualElement.Arrange\(Rect\)](#), [VisualElement.Layout\(Rect\)](#),  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#),  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#), [VisualElement.Visual](#),  
[VisualElement.FlowDirection](#), [VisualElement.Window](#), [VisualElement.AnchorX](#),

[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsFocused](#) ,  
[VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## MultiBarChart()

```
public MultiBarChart()
```

# Fields

## AutoCalculateItemSeparationMarginProperty

```
public static readonly BindableProperty AutoCalculateItemSeparationMarginProperty
```

Field Value

[BindableProperty](#)

## BarsCornerRadiusProperty

```
public static readonly BindableProperty BarsCornerRadiusProperty
```

Field Value

[BindableProperty](#)

## ColumnNamesProperty

```
public static readonly BindableProperty ColumnNamesProperty
```

Field Value

[BindableProperty](#)

## EntriesProperty

```
public static readonly BindableProperty EntriesProperty
```

Field Value

[BindableProperty](#)

## GroupStylesProperty

```
public static readonly BindableProperty GroupStylesProperty
```

Field Value

[BindableProperty](#)

## ItemSeparationMarginProperty

```
public static readonly BindableProperty ItemSeparationMarginProperty
```

Field Value

[BindableProperty](#)

## Properties

### AutoCalculateItemSeparationMargin

Gets or sets the separation margin between each item. Default is 8

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

Property Value

[bool](#)

## BarsCornerRadius

```
public float BarsCornerRadius { get; set; }
```

### Property Value

[float](#)

## ColumnNames

```
public ObservableCollection<string> ColumnNames { get; set; }
```

### Property Value

[ObservableCollection](#)<[string](#)>

## Entries

```
public ObservableCollection<ChartItem> Entries { get; set; }
```

### Property Value

[ObservableCollection](#)<[ChartItem](#)>

## GroupStyles

```
public ObservableCollection<ChartGroupStyle> GroupStyles { get; set; }
```

### Property Value

[ObservableCollection](#)<[ChartGroupStyle](#)>

## ItemSeparationMargin

Gets or sets the separation margin between each item. Default is 8

```
public float ItemSeparationMargin { get; set; }
```

Property Value

[float](#) ↗

# Class MultiBarChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public sealed class MultiBarChartDrawable : BaseChartDrawable, IDrawable, IDisposable
```

## Inheritance

[object](#) ← [BaseChartDrawable](#) ← MultiBarChartDrawable

## Implements

[IDisposable](#), [IDisposable](#)

## Inherited Members

[BaseChartDrawable.Initialized](#), [BaseChartDrawable.IsAnimating](#), [BaseChartDrawable.ColorBrush](#),  
[BaseChartDrawable.AxisDashPattern](#), [BaseChartDrawable.DisplayHeaderValues](#),  
[BaseChartDrawable.PathsColorOpacity](#), [BaseChartDrawable.DisplayHorizontalAxisLines](#),  
[BaseChartDrawable.DisplayVerticalAxisLines](#), [BaseChartDrawable.AnimationProgress](#),  
[BaseChartDrawable.LabelTextTruncationEnabled](#), [BaseChartDrawable.DisplayValueLabelsOnTop](#),  
[BaseChartDrawable.StrokeSize](#), [BaseChartDrawable.AxisLinesStrokeSize](#),  
[BaseChartDrawable.AxisLinesColor](#), [BaseChartDrawable.StrokeColor](#),  
[BaseChartDrawable.ItemSeparationMargin](#), [BaseChartDrawable.FooterLabelsTextSize](#),  
[BaseChartDrawable.FooterLabelsMargin](#), [BaseChartDrawable.HeaderValuesMargin](#),  
[BaseChartDrawable.FillColor](#), [BaseChartDrawable.Font](#), [BaseChartDrawable.Entries](#),  
[BaseChartDrawable.FontSize](#), [BaseChartDrawable.AxisFontSize](#), [BaseChartDrawable.Margin](#),  
[BaseChartDrawable.FontColor](#), [BaseChartDrawable.FooterLabelsFontColor](#),  
[BaseChartDrawable.EnableAntialias](#), [BaseChartDrawable.Draw\(ICanvas, RectF\)](#),  
[BaseChartDrawable.ClearBackground\(ICanvas, SizeF, Color\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Properties

### AutoCalculateItemSeparationMargin

Indicates if control will calculate by itself left and right bar margins based on the group entries.

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

Property Value

[bool](#)

## BarsCornerRadius

Border radius to use when drawing each bar

```
public float BarsCornerRadius { get; set; }
```

Property Value

[float](#)

## BarsFillColor

Gets or sets the color to use when drawing each bar

```
public Color BarsFillColor { get; set; }
```

Property Value

[Color](#)

## ColumnNames

```
public ObservableCollection<string> ColumnNames { get; set; }
```

Property Value

[ObservableCollection](#)<[string](#)>

## GroupStyles

```
public ObservableCollection<ChartGroupStyle> GroupStyles { get; set; }
```

## Property Value

[ObservableCollection](#) <[ChartGroupStyle](#)>

## Methods

### Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public override void Dispose()
```

### DrawChart(ICanvas, RectF)

Calculates and draws the current chart with the Entries[] data

```
public override void DrawChart(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

**canvas** [ICanvas](#)

Current canvas

**dirtyRect** [RectF](#)

Current canvas rect

### DrawVerticalStepLines(ICanvas, PointF[], float, float)

```
protected override void DrawVerticalStepLines(ICanvas canvas, PointF[] points, float origin, float maxPositiveValue)
```

Parameters

canvas [ICanvas](#)

points [PointF](#)[]

origin [float](#)

maxPositiveValue [float](#)

## Entry\_PropertyChanged(object, PropertyChangedEventArgs)

```
protected override void Entry_PropertyChanged(object sender, PropertyChangedEventArgs e)
```

Parameters

sender [object](#)

e [PropertyChangedEventArgs](#)

# Class MultiLineChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class MultiLineChartDrawable : BaseChartDrawable, IDrawable, IDisposable
```

## Inheritance

[object](#) ← [BaseChartDrawable](#) ← MultiLineChartDrawable

## Implements

[IDisposable](#), [IDisposable](#)

## Inherited Members

[BaseChartDrawable.canvas](#), [BaseChartDrawable.displayHeaderValues](#),  
[BaseChartDrawable.MaxValueCoordinate](#), [BaseChartDrawable.AxisXMargin](#),  
[BaseChartDrawable.dirtyRect](#), [BaseChartDrawable.displayValueLabelsOnTop](#),  
[BaseChartDrawable.colorBrush](#), [BaseChartDrawable.Initialized](#), [BaseChartDrawable.IsAnimating](#),  
[BaseChartDrawable.ColorBrush](#), [BaseChartDrawable.AxisDashPattern](#),  
[BaseChartDrawable.DisplayHeaderValues](#), [BaseChartDrawable.PathsColorOpacity](#),  
[BaseChartDrawable.DisplayHorizontalAxisLines](#), [BaseChartDrawable.DisplayVerticalAxisLines](#),  
[BaseChartDrawable.AnimationProgress](#), [BaseChartDrawable.LabelTextTruncationEnabled](#),  
[BaseChartDrawable.DisplayValueLabelsOnTop](#), [BaseChartDrawable.StrokeSize](#),  
[BaseChartDrawable.AxisLinesStrokeSize](#), [BaseChartDrawable.AxisLinesColor](#),  
[BaseChartDrawable.StrokeColor](#), [BaseChartDrawable.ItemSeparationMargin](#),  
[BaseChartDrawable.FooterLabelsTextSize](#), [BaseChartDrawable.FooterLabelsMargin](#),  
[BaseChartDrawable.HeaderValuesMargin](#), [BaseChartDrawable.FillColor](#), [BaseChartDrawable.Font](#),  
[BaseChartDrawable.Entries](#),  
[BaseChartDrawable.Entries.CollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#),  
[BaseChartDrawable.FontSize](#), [BaseChartDrawable.AxisFontSize](#), [BaseChartDrawable.Margin](#),  
[BaseChartDrawable.FontColor](#), [BaseChartDrawable.FooterLabelsFontColor](#),  
[BaseChartDrawable.EnableAntialias](#), [BaseChartDrawable.Points](#), [BaseChartDrawable.ItemSize](#),  
[BaseChartDrawable.MinValue](#), [BaseChartDrawable.MaxValue](#), [BaseChartDrawable.Width](#),  
[BaseChartDrawable.Height](#), [BaseChartDrawable.Origin](#), [BaseChartDrawable.HeaderHeight](#),  
[BaseChartDrawable.FooterHeight](#), [BaseChartDrawable.ValueRange](#),  
[BaseChartDrawable.RequestInvalidate\(\)](#),  
[BaseChartDrawable.CalculatePaddingForLabelValue\(float, SizeF\)](#),  
[BaseChartDrawable.CalculateHorizontalYLineCoordinate\(float, SizeF, float, float\)](#),

[BaseChartDrawable.DrawHorizontalStepLines\(ICanvas, IEnumerable<PointF>, float, float, float, float\)](#),  
[BaseChartDrawable.Draw\(ICanvas, RectF\)](#), [BaseChartDrawable.MeasureLabels\(\)](#),  
[BaseChartDrawable.CalculateFooterHeight\(SizeF\[\]\)](#), [BaseChartDrawable.CalculateHeaderHeight\(SizeF\[\]\)](#),  
[BaseChartDrawable.CalculateItemSize\(float, float, float, float\)](#),  
[BaseChartDrawable.CalculateItemSize\(List<ChartItem>, float, float, float, float\)](#),  
[BaseChartDrawable.ClearBackground\(ICanvas, SizeF, Color\)](#),  
[BaseChartDrawable.CalculateYOrigin\(float, float\)](#), [BaseChartDrawable.CalculatePoints\(SizeF, float, float\)](#),  
[BaseChartDrawable.CalculatePoints\(List<ChartItem>, SizeF, float, float\)](#),  
[BaseChartDrawable.DrawLabels\(List<ChartItem>, ICanvas, PointF\[\], SizeF, float, float, float\)](#),  
[BaseChartDrawable.DrawLabels\(ICanvas, PointF\[\], SizeF, float, float, float\)](#),  
[BaseChartDrawable.TruncateText\(ICanvas, SizeF, string\)](#), [object.Equals\(object\)](#) ↗,  
[object.Equals\(object, object\)](#) ↗, [object.GetHashCode\(\)](#) ↗, [object.GetType\(\)](#) ↗,  
[object.MemberwiseClone\(\)](#) ↗, [object.ReferenceEquals\(object, object\)](#) ↗, [object.ToString\(\)](#) ↗

## Properties

### ColumnNames

```
public ObservableCollection<string> ColumnNames { get; set; }
```

### Property Value

[ObservableCollection](#) ↗ <[string](#) ↗>

### CurveFactor

Sets how 'curvy' the bezier curve will be when drawn. Accepts values between 0-1. Default is 0.6

```
public float CurveFactor { get; set; }
```

### Property Value

[float](#) ↗

### ExpandAndFillBackgroundCurvePath

If true Bezier curve Background Path color will expand and fill when drawn.

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

Property Value

[bool](#)

## FillCurveColor

Gets or sets the color to use when filling Curve serie

```
public Color FillCurveColor { get; set; }
```

Property Value

[Color](#)

## GroupStyles

```
public ObservableCollection<ChartGroupStyle> GroupStyles { get; set; }
```

Property Value

[ObservableCollection](#)<[ChartGroupStyle](#)>

## IsCurveBackgroundFilled

Gets or sets if the curve will have a solid color background when drawn.

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

Property Value

[bool](#)

## LineColor

Gets or sets the color to use when drawing chart lines.

```
public Color LineColor { get; set; }
```

### Property Value

[Color](#)

## PointSize

Gets or sets the radius for each point.

```
public float PointSize { get; set; }
```

### Property Value

[float](#)

## ShowPointsForCurveStyle

If true and Style prop equals to 'Curve' points will be shown when drawing cubic bezier

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

### Property Value

[bool](#)

## Style

Defines the style for the current LineChart

```
public ChartEnums.LineChartStyle Style { get; set; }
```

## Property Value

[ChartEnums.LineChartStyle](#)

## Methods

### Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public override void Dispose()
```

### DrawChart(ICanvas, RectF)

Calculates and draws the current chart with the Entries[] data

```
public override void DrawChart(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

Current canvas

dirtyRect [RectF](#)

Current canvas rect

### DrawVerticalStepLines(ICanvas, PointF[], float, float)

```
protected override void DrawVerticalStepLines(ICanvas canvas, PointF[] points, float origin,
float maxPositiveValue)
```

#### Parameters

canvas [ICanvas](#)

points [PointF](#)[]

origin [float](#)

maxPositiveValue [float](#)

## Entry\_PropertyChanged(object, PropertyChangedEventArgs)

```
protected override void Entry_PropertyChanged(object sender, PropertyChangedEventArgs e)
```

### Parameters

sender [object](#)

e [PropertyChangedEventArgs](#)

# Class MultiLineChartView

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The MultiLineChartView is a drawn control for displaying multiple line charts within a single view. This enables the visualization of multiple data series, each represented by a separate line, on a shared coordinate system.

The class is derived from BaseChart, allowing it to inherit essential charting properties and behaviors while adding specialized features for multi-line visualization.

```
public sealed class MultiLineChartView : BaseChart, INotifyPropertyChanged,  
    IVisualTreeElement, IEffectorControlProvider, IToolTipElement, IContextFlyoutElement,  
    IAnimatable, IViewController, IVisualElementController, IElementController,  
    IGestureController, IGestureRecognizers, IPropertyMapperView, IHotReloadableView,  
    IReplaceableView, IGraphicsView, IView, IElement, ITransform, IDisposable
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← [BaseChart](#) ← MultiLineChartView

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectorControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#),  
[IDisposable](#)

## Inherited Members

[BaseChart.EasingProperty](#), [BaseChart.Easing](#), [BaseChart.BackgroundProperty](#), [BaseChart.Background](#),  
[BaseChart.AxisDashPatternProperty](#), [BaseChart.AxisDashPattern](#),  
[BaseChart.DisplayHeaderValuesProperty](#), [BaseChart.DisplayHeaderValues](#),  
[BaseChart.PathsColorOpacityProperty](#), [BaseChart.PathsColorOpacity](#),  
[BaseChart.AxisLinesFontSizeProperty](#), [BaseChart.AxisLinesFontSize](#), [BaseChart.AxisLineColorProperty](#),  
[BaseChart.AxisLinesColor](#), [BaseChart.AxisLinesStrokeSizeProperty](#), [BaseChart.AxisLinesStrokeSize](#),  
[BaseChart.DisplayVerticalAxisLinesProperty](#), [BaseChart.DisplayVerticalAxisLines](#),  
[BaseChart.DisplayHorizontalAxisLinesProperty](#), [BaseChart.DisplayHorizontalAxisLines](#),  
[BaseChart.AnimationIntervalProperty](#), [BaseChart.AnimationInterval](#),  
[BaseChart.ReanimateOnPropertyChangedProperty](#), [BaseChart.ReanimateOnPropertyChanged](#),

[BaseChart.EnableEntryAnimationsProperty](#), [BaseChart.EnableEntryAnimations](#),  
[BaseChart.IsLabelTextTruncationEnabledProperty](#), [BaseChart.IsLabelTextTruncationEnabled](#),  
[BaseChart.DisplayValueLabelsOnTopProperty](#), [BaseChart.DisplayValueLabelsOnTop](#),  
[BaseChart.StrokeSizeProperty](#), [BaseChart.StrokeSize](#), [BaseChart.StrokeColorProperty](#),  
[BaseChart.StrokeColor](#), [BaseChart.ItemSeparationMarginProperty](#), [BaseChart.ItemSeparationMargin](#),  
[BaseChart.FooterLabelsTextSizeProperty](#), [BaseChart.FooterLabelsTextSize](#),  
[BaseChart.FooterLabelsMarginProperty](#), [BaseChart.FooterLabelsMargin](#),  
[BaseChart.HeaderValuesMarginProperty](#), [BaseChart.HeaderValuesMargin](#), [BaseChart.FillColorProperty](#),  
[BaseChart.FillColor](#), [BaseChart.FontFamilyProperty](#), [BaseChart.FontFamily](#), [BaseChart.FontSizeProperty](#),  
[BaseChart.FontSize](#), [BaseChart.ChartMarginProperty](#), [BaseChart.ChartMargin](#),  
[BaseChart.FontColorProperty](#), [BaseChart.FontColor](#), [BaseChart.FooterLabelsFontColorProperty](#),  
[BaseChart.FooterLabelsFontColor](#), [BaseChart.EnableAntialiasProperty](#), [BaseChart.EnableAntialias](#),  
[BaseChart.Dispose\(\)](#), [GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#),  
[GraphicsView.Drawable](#), [GraphicsView.StartHoverInteraction](#),  
[GraphicsView.MoveHoverInteraction](#), [GraphicsView.EndHoverInteraction](#),  
[GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#), [GraphicsView.EndInteraction](#),  
[GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#), [View.HorizontalOptionsProperty](#),  
[View.MarginProperty](#), [View.GetChildElements\(Point\)](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),  
[VisualElement.VisualProperty](#), [VisualElement.IsVisibleProperty](#), [VisualElement.OpacityProperty](#),  
[VisualElement.BackgroundColorProperty](#), [VisualElement.BehaviorsProperty](#),  
[VisualElement.TriggersProperty](#), [VisualElement.WidthRequestProperty](#),  
[VisualElement.HeightRequestProperty](#), [VisualElement.MinimumWidthRequestProperty](#),  
[VisualElement.MinimumHeightRequestProperty](#), [VisualElement.MaximumWidthRequestProperty](#),  
[VisualElement.MaximumHeightRequestProperty](#), [VisualElement.IsFocusedProperty](#),  
[VisualElement.FlowDirectionProperty](#), [VisualElement.WindowProperty](#),  
[VisualElement.ShadowProperty](#), [VisualElement.ZIndexProperty](#),  
[VisualElement.ControlsVisualElementMapper](#), [VisualElement.BatchBegin\(\)](#),  
[VisualElement.BatchCommit\(\)](#), [VisualElement.Focus\(\)](#),  
[VisualElement.Measure\(double, double, MeasureFlags\)](#), [VisualElement.Unfocus\(\)](#),  
[VisualElement.Arrange\(Rect\)](#), [VisualElement.Layout\(Rect\)](#),  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#),  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#), [VisualElement.Visual](#),

[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsFocused](#) ,  
[VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## MultiLineChartView()

```
public MultiLineChartView()
```

# Fields

## ChartStyleProperty

```
public static readonly BindableProperty ChartStyleProperty
```

Field Value

[BindableProperty](#)

## ColumnNamesProperty

```
public static readonly BindableProperty ColumnNamesProperty
```

Field Value

[BindableProperty](#)

## CurveFactorProperty

```
public static readonly BindableProperty CurveFactorProperty
```

Field Value

[BindableProperty](#)

## EntriesProperty

```
public static readonly BindableProperty EntriesProperty
```

Field Value

[BindableProperty](#)

## ExpandAndFillBackgroundCurvePathProperty

```
public static readonly BindableProperty ExpandAndFillBackgroundCurvePathProperty
```

Field Value

[BindableProperty](#)

## GroupStylesProperty

```
public static readonly BindableProperty GroupStylesProperty
```

Field Value

[BindableProperty](#)

## IsCurveBackgroundFilledProperty

```
public static readonly BindableProperty IsCurveBackgroundFilledProperty
```

Field Value

[BindableProperty](#)

## PointSizeProperty

```
public static readonly BindableProperty PointSizeProperty
```

Field Value

[BindableProperty](#)

## ShowPointsForCurveStyleProperty

```
public static readonly BindableProperty ShowPointsForCurveStyleProperty
```

Field Value

[BindableProperty](#)

## Properties

### ChartStyle

Defines the style for the current LineChart

```
public ChartEnums.LineChartStyle ChartStyle { get; set; }
```

Property Value

[ChartEnums.LineChartStyle](#)

### ColumnNames

```
public ObservableCollection<string> ColumnNames { get; set; }
```

Property Value

[ObservableCollection](#)<[string](#)>

## CurveFactor

Sets how 'curvy' the bezier curve will be when drawn. Accepts values between 0-1. Default is 0.6

```
public float CurveFactor { get; set; }
```

Property Value

[float](#)

## Entries

```
public ObservableCollection<ChartItem> Entries { get; set; }
```

Property Value

[ObservableCollection](#)<[ChartItem](#)>

## ExpandAndFillBackgroundCurvePath

If true Bezier curve Background Path color will expand and fill when drawn.

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

Property Value

[bool](#)

## GroupStyles

```
public ObservableCollection<ChartGroupStyle> GroupStyles { get; set; }
```

Property Value

[ObservableCollection](#)<[ChartGroupStyle](#)>

## IsCurveBackgroundFilled

Gets or sets if the curve path will have a solid color background when drawn.

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

Property Value

[bool](#)

## PointSize

Gets or sets the radius for each point.

```
public float PointSize { get; set; }
```

Property Value

[float](#)

## ShowPointsForCurveStyle

If true and Style prop equals to 'Curve' points will be shown when drawing cubic bezier

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

Property Value

[bool](#)

# Class NumericUpDown

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The NumericUpDown is a drawn control for selecting numeric values by incrementing or decrementing them with interactive buttons or input.

```
public class NumericUpDown : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← NumericUpDown

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),

[VisualElement.VisualProperty](#) , [VisualElement.VisibileProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,

[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### NumericUpDown()

```
public NumericUpDown()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

[BindableProperty](#)

## ColorProperty

```
public static readonly BindableProperty ColorProperty
```

Field Value

[BindableProperty](#) ↗

## FontSizeProperty

```
public static readonly BindableProperty FontSizeProperty
```

Field Value

[BindableProperty](#) ↗

## IntervalProperty

```
public static readonly BindableProperty IntervalProperty
```

Field Value

[BindableProperty](#) ↗

## MaximumColorProperty

```
public static readonly BindableProperty MaximumColorProperty
```

Field Value

[BindableProperty](#) ↗

## MaximumProperty

```
public static readonly BindableProperty MaximumProperty
```

Field Value

[BindableProperty ↗](#)

## MaximumTextColorProperty

```
public static readonly BindableProperty MaximumTextColorProperty
```

Field Value

[BindableProperty ↗](#)

## MinimumColorProperty

```
public static readonly BindableProperty MinimumColorProperty
```

Field Value

[BindableProperty ↗](#)

## MinimumProperty

```
public static readonly BindableProperty MinimumProperty
```

Field Value

[BindableProperty ↗](#)

## MinimumTextColorProperty

```
public static readonly BindableProperty MinimumTextColorProperty
```

Field Value

[BindableProperty](#)

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty](#)

## ValueProperty

```
public static readonly BindableProperty ValueProperty
```

Field Value

[BindableProperty](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

### Color

```
public Color Color { get; set; }
```

Property Value

[Color](#)

## FontSize

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

Property Value

[double](#)

## Interval

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

Property Value

[double](#)

## Maximum

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

Property Value

[double](#)

## MaximumColor

```
public Brush MaximumColor { get; set; }
```

Property Value

[Brush](#)

## MaximumTextColor

```
public Color MaximumTextColor { get; set; }
```

Property Value

[Color](#)

## Minimum

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

Property Value

[double](#)

## MinimumColor

```
public Brush MinimumColor { get; set; }
```

Property Value

[Brush](#)

## MinimumTextColor

```
public Color MinimumTextColor { get; set; }
```

Property Value

[Color](#) ↗

## NumericUpDownDrawable

```
public NumericUpDownDrawable NumericUpDownDrawable { get; set; }
```

Property Value

[NumericUpDownDrawable](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#) ↗

## Value

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

Property Value

[double](#) ↗

## Methods

OnParentSet()

Raises the (internal) **ParentSet** event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet\(\)](#)

## Events

### ValueChanged

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

#### Event Type

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

# Class NumericUpDownDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class NumericUpDownDrawable : IDrawable
```

## Inheritance

[object](#) ← NumericUpDownDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## Color

```
public Color Color { get; set; }
```

### Property Value

[Color](#)

## FontSize

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

Property Value

[double](#)

## Maximum

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

Property Value

[double](#)

## MaximumColorPaint

```
public Paint MaximumColorPaint { get; set; }
```

Property Value

[Paint](#)

## MaximumTextColor

```
public Color MaximumTextColor { get; set; }
```

Property Value

[Color](#)

## Minimum

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

Property Value

[double](#) ↗

## MinimumColorPaint

```
public Paint MinimumColorPaint { get; set; }
```

Property Value

[Paint](#) ↗

## MinimumTextColor

```
public Color MinimumTextColor { get; set; }
```

Property Value

[Color](#) ↗

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#) ↗

## Value

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

Property Value

[double](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBorder(ICanvas, RectF)

```
public virtual void DrawBorder(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawMaximum(ICanvas, RectF)

```
public virtual void DrawMaximum(ICanvas canvas, RectF dirtyRect)
```

Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawMinus(ICanvas, RectF)

```
public virtual void DrawMinus(ICanvas canvas, RectF dirtyRect)
```

Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawValue(ICanvas, RectF)

```
public virtual void DrawValue(ICanvas canvas, RectF dirtyRect)
```

Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

# Class PieChart

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The PieChart is a drawn control for rendering pie charts to visually represent data as portions of a circle.

```
public class PieChart : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← PieChart
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
VisualElement.ScaleXProperty ↩ , VisualElement.ScaleYProperty ↩ , VisualElement.ClipProperty ↩ ,
VisualElement.VisualProperty ↩ , VisualElement.IsVisibleProperty ↩ , VisualElement.OpacityProperty ↩ ,
```

[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,

[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### PieChart()

```
public PieChart()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

### BindableProperty

## ItemsSourceProperty

```
public static readonly BindableProperty ItemsSourceProperty
```

Field Value

[BindableProperty](#)

## ShowLabelsProperty

```
public static readonly BindableProperty ShowLabelsProperty
```

Field Value

[BindableProperty](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

### ItemsSource

```
public Dictionary<string, float> ItemsSource { get; set; }
```

Property Value

[Dictionary](#)<[string](#), [float](#)>

## PieChartDrawable

```
public PieChartDrawable PieChartDrawable { get; set; }
```

### Property Value

[PieChartDrawable](#)

## ShowLabels

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

### Property Value

[bool](#)

## Methods

### OnParentSet()

Raises the (internal) [ParentSet](#) event.

```
protected override void OnParentSet()
```

### Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

### See Also

[OnParentSet\(\)](#)

# Class PieChartDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class PieChartDrawable : IDrawable
```

## Inheritance

[object](#) ← PieChartDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

## Property Value

[Paint](#)

## ItemsSource

```
public Dictionary<string, float> ItemsSource { get; set; }
```

## Property Value

[Dictionary](#)<[string](#), [float](#)>

## ShowLabels

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

### Property Value

[bool](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

[canvas](#) [ICanvas](#)

[dirtyRect](#) [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

[canvas](#) [ICanvas](#)

[dirtyRect](#) [RectF](#)

### DrawLabels(ICanvas, RectF)

```
public virtual void DrawLabels(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawSlices(ICanvas, RectF)

```
public virtual void DrawSlices(ICanvas canvas, RectF dirtyRect)
```

### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

# Class ProgressBar

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ProgressBar class represents a customizable control for displaying the progress of a task visually.

```
public class ProgressBar : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ← [GraphicsView](#) ← [ProgressBar](#)

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),  
[VisualElement.VisualProperty](#), [VisualElement.IsVisibleProperty](#), [VisualElement.OpacityProperty](#),

[VisualElement.BackgroundColorProperty](#) , [VisualElement.BackgroundProperty](#) ,  
[VisualElement.BehaviorsProperty](#) , [VisualElement.TriggersProperty](#) ,  
[VisualElement.WidthRequestProperty](#) , [VisualElement.HeightRequestProperty](#) ,  
[VisualElement.MinimumWidthRequestProperty](#) , [VisualElement.MinimumHeightRequestProperty](#) ,  
[VisualElement.MaximumWidthRequestProperty](#) , [VisualElement.MaximumHeightRequestProperty](#) ,  
[VisualElement.IsFocusedProperty](#) , [VisualElement.FlowDirectionProperty](#) ,  
[VisualElement.WindowProperty](#) , [VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Background](#) ,  
[VisualElement.Behaviors](#) , [VisualElement.Bounds](#) , [VisualElement.Height](#) ,  
[VisualElement.HeightRequest](#) , [VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) ,  
[VisualElement.IsEnabledCore](#) , [VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) ,  
[VisualElement.MinimumHeightRequest](#) , [VisualElement.MinimumWidthRequest](#) ,  
[VisualElement.MaximumHeightRequest](#) , [VisualElement.MaximumWidthRequest](#) ,  
[VisualElement.Opacity](#) , [VisualElement.Rotation](#) , [VisualElement.RotationX](#) ,  
[VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) , [VisualElement.ScaleY](#) ,  
[VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) , [VisualElement.Triggers](#) ,  
[VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) , [VisualElement.Y](#) ,  
[VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) , [VisualElement.Handler](#) ,  
[VisualElement.Shadow](#) , [VisualElement.ZIndex](#) , [VisualElement.DesiredSize](#) ,  
[VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) , [VisualElement.Focused](#) ,  
[VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) , [VisualElement.Unfocused](#) ,  
[VisualElement.Loaded](#) , [VisualElement.Unloaded](#) , [NavigableElement.OnParentSet\(\)](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,

[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesIsInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### ProgressBar()

```
public ProgressBar()
```

## Fields

### CornerRadiusProperty

```
public static readonly BindableProperty CornerRadiusProperty
```

## Field Value

[BindableProperty](#)

## EasingIntervalProperty

```
public static readonly BindableProperty EasingIntervalProperty
```

Field Value

[BindableProperty](#) ↗

## EasingProperty

```
public static readonly BindableProperty EasingProperty
```

Field Value

[BindableProperty](#) ↗

## EnableAnimationsProperty

```
public static readonly BindableProperty EnableAnimationsProperty
```

Field Value

[BindableProperty](#) ↗

## IsVerticalProperty

```
public static readonly BindableProperty IsVerticalProperty
```

Field Value

[BindableProperty](#) ↗

## ProgressBrushProperty

```
public static readonly BindableProperty ProgressBrushProperty
```

Field Value

[BindableProperty↗](#)

## ProgressProperty

```
public static readonly BindableProperty ProgressProperty
```

Field Value

[BindableProperty↗](#)

## RoundCornersProperty

```
public static readonly BindableProperty RoundCornersProperty
```

Field Value

[BindableProperty↗](#)

## StrokeBrushProperty

```
public static readonly BindableProperty StrokeBrushProperty
```

Field Value

[BindableProperty↗](#)

## Properties

## CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

Property Value

[CornerRadius](#) ↗

## Easing

```
public Easing Easing { get; set; }
```

Property Value

[Easing](#) ↗

## EasingInterval

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

Property Value

[int](#) ↗

## EnableAnimations

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

Property Value

[bool](#) ↗

## IsVertical

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

Property Value

[bool](#)

## Progress

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

Property Value

[double](#)

## ProgressBarDrawable

```
protected ProgressBarDrawable ProgressBarDrawable { get; set; }
```

Property Value

[ProgressBarDrawable](#)

## ProgressBrush

```
public Brush ProgressBrush { get; set; }
```

Property Value

[Brush](#)

## RoundCorners

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

Property Value

[bool](#)

## StrokeBrush

```
public Brush StrokeBrush { get; set; }
```

Property Value

[Brush](#)

## Methods

### OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

### Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

### UpdateCornerRadius()

```
protected void UpdateCornerRadius()
```

### UpdateProgress()

```
protected void UpdateProgress()
```

## UpdateRoundCorners()

```
protected void UpdateRoundCorners()
```

## Events

### ValueChanged

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

#### Event Type

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

# Class ProgressBarDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class ProgressBarDrawable : IDrawable
```

## Inheritance

[object](#) ← ProgressBarDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## CornerRadius

```
public CornerRadius CornerRadius { get; set; }
```

### Property Value

[CornerRadius](#)

## IsAnimating

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

### Property Value

[bool](#)

## IsVertical

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

Property Value

[bool](#)

## Progress

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

Property Value

[double](#)

## ProgressPaint

```
public Paint ProgressPaint { get; set; }
```

Property Value

[Paint](#)

## StrokePaint

```
public Paint StrokePaint { get; set; }
```

Property Value

[Paint](#)

## Style

```
public ProgressBarStyle Style { get; set; }
```

## Property Value

[ProgressBarStyle](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawProgress(ICanvas, RectF)

```
public virtual void DrawProgress(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawTrack(ICanvas, RectF)

```
public virtual void DrawTrack(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

`dirtyRect` [RectF](#)

# Enum ProgressBarStyle

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ProgressBarStyle enumeration defines the visual style options for rendering a progress bar. This provides developers with a simple way to specify whether the progress bar appears with sharp corners or rounded edges.

```
public enum ProgressBarStyle
```

## Fields

Rounded = 1

Square = 0

# Class ProgressRadial

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ProgressRadial is a drawn control designed to visually represent progress in a circular or radial form. It can be used to display percentages or completion statuses, such as task progress, loading indicators, or performance metrics.

```
public class ProgressRadial : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← ProgressRadial
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
```

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.Background](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.OnParentSet\(\)](#) , [NavigableElement.Navigation](#) , [NavigableElement.Style](#) ,  
[NavigableElement.StyleClass](#) , [NavigableElement.class](#) , [Element.AutomationIdProperty](#) ,  
[Element.ClassIdProperty](#) , [Element.ControlsElementMapper](#) ,  
[Element.InsertLogicalChild\(int, Element\)](#) , [Element.AddLogicalChild\(Element\)](#) ,  
[Element.RemoveLogicalChild\(Element\)](#) , [Element.ClearLogicalChildren\(\)](#) ,

[Element.FindByName\(string\)](#) , [Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### ProgressRadial()

```
public ProgressRadial()
```

## Fields

### BackgroundColorProperty

```
public static readonly BindableProperty BackgroundColorProperty
```

## Field Value

[BindableProperty](#)

## DirectionProperty

```
public static readonly BindableProperty DirectionProperty
```

Field Value

[BindableProperty](#)

## FontSizeProperty

```
public static readonly BindableProperty FontSizeProperty
```

Field Value

[BindableProperty](#)

## MaximumProperty

```
public static readonly BindableProperty MaximumProperty
```

Field Value

[BindableProperty](#)

## MinimumProperty

```
public static readonly BindableProperty MinimumProperty
```

Field Value

[BindableProperty](#)

## ProgressColorProperty

```
public static readonly BindableProperty ProgressColorProperty
```

Field Value

[BindableProperty ↗](#)

## StrokeColorProperty

```
public static readonly BindableProperty StrokeColorProperty
```

Field Value

[BindableProperty ↗](#)

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty ↗](#)

## ValueProperty

```
public static readonly BindableProperty ValueProperty
```

Field Value

[BindableProperty ↗](#)

## Properties

## BackgroundColor

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

### Property Value

[Color](#)

## Direction

```
public ProgressRadialDirection Direction { get; set; }
```

### Property Value

[ProgressRadialDirection](#)

## FontSize

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

### Property Value

[double](#)

## Maximum

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

### Property Value

[int](#)

## Minimum

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

Property Value

[int ↗](#)

## ProgressColor

```
public Color ProgressColor { get; set; }
```

Property Value

[Color ↗](#)

## StrokeColor

```
public Color StrokeColor { get; set; }
```

Property Value

[Color ↗](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color ↗](#)

## Value

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

Property Value

[int](#)

## Methods

### OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

## Events

### ValueChanged

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

Event Type

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

# Enum ProgressRadialDirection

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ProgressRadialDirection enumeration specifies the direction of progress rendering for radial progress components, allowing developers to define whether progress moves clockwise or counterclockwise along the circular axis.

```
public enum ProgressRadialDirection
```

## Fields

**LeftToRight** = 0

**RightToLeft** = 1

# Class ProgressRadialDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class ProgressRadialDrawable : IDrawable
```

## Inheritance

[object](#) ← ProgressRadialDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundColor

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

## Property Value

[Color](#)

## Direction

```
public ProgressRadialDirection Direction { get; set; }
```

## Property Value

[ProgressRadialDirection](#)

## FontSize

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

Property Value

[double](#)

## IsAnimating

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

Property Value

[bool](#)

## ProgressAngle

```
public float ProgressAngle { get; set; }
```

Property Value

[float](#)

## ProgressColor

```
public Color ProgressColor { get; set; }
```

Property Value

[Color](#)

## ProgressText

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

Property Value

[string](#)

## StrokeColor

```
public Color StrokeColor { get; set; }
```

Property Value

[Color](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawProgress(ICanvas, RectF)

```
public virtual void DrawProgress(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawStroke(ICanvas, RectF)

```
public virtual void DrawStroke(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawText(ICanvas, RectF)

```
public virtual void DrawText(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Class Pulselcon

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Pulselcon is a drawn control that creates an animated pulsing effect around an icon. This control is particularly useful for drawing attention to specific elements in your user interface, such as notifications, interactive features, or status indicators.

```
public class PulseIcon : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← Pulselcon

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.OnParentSet\(\)](#) , [NavigableElement.Navigation](#) , [NavigableElement.Style](#) ,  
[NavigableElement.StyleClass](#) , [NavigableElement.class](#) , [Element.AutomationIdProperty](#) ,  
[Element.ClassIdProperty](#) , [Element.ControlsElementMapper](#) ,  
[Element.InsertLogicalChild\(int, Element\)](#) , [Element.AddLogicalChild\(Element\)](#) ,  
[Element.RemoveLogicalChild\(Element\)](#) , [Element.ClearLogicalChildren\(\)](#) ,

[Element.FindByName\(string\)](#) , [Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### PulseIcon()

```
public PulseIcon()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

[BindableProperty](#)

## IsPulsingProperty

```
public static readonly BindableProperty IsPulsingProperty
```

Field Value

[BindableProperty](#)

## PulseColorProperty

```
public static readonly BindableProperty PulseColorProperty
```

Field Value

[BindableProperty](#)

## SourceProperty

```
public static readonly BindableProperty SourceProperty
```

Field Value

[BindableProperty](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

## IsPulsing

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

Property Value

[bool](#)

## PulseColor

```
public Color PulseColor { get; set; }
```

Property Value

[Color](#)

## PulseIconDrawable

```
public PulseIconDrawable PulseIconDrawable { get; set; }
```

Property Value

[PulseIconDrawable](#)

## Source

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

Property Value

[string](#)

# Methods

## OnParentChanged()

When overridden in a derived class, should raise the [ParentChanged](#) event.

```
protected override void OnParentChanged()
```

### Remarks

It is the implementor's responsibility to raise the [ParentChanged](#) event.

# Events

## Clicked

```
public event EventHandler Clicked
```

### Event Type

[EventHandler](#)

# Class PulseIconDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class PulseIconDrawable : IDrawable
```

## Inheritance

[object](#) ← PulseIconDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

## Properties

### BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

#### Property Value

[Paint](#)

### PulseColor

```
public Color PulseColor { get; set; }
```

#### Property Value

[Color](#)

## Source

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

## Property Value

[string](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawIcon(ICanvas, RectF)

```
public virtual void DrawIcon(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawPulse(ICanvas, RectF)

```
public virtual void DrawPulse(ICanvas canvas, RectF dirtyRect)
```

### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

# Class Rating

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Rating is a drawn control that allows users to rate items or entities, typically using a star-based system or other graphical representation. This control is perfect for scenarios where users need to provide feedback, score performance, or indicate preferences in an intuitive visual format.

```
public class Rating : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← Rating
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
```

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### Rating()

```
public Rating()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

### Field Value

[BindableProperty](#)

## IsReadOnlyProperty

```
public static readonly BindableProperty IsReadOnlyProperty
```

Field Value

[BindableProperty](#)

## ItemCountProperty

```
public static readonly BindableProperty ItemCountProperty
```

Field Value

[BindableProperty](#)

## SelectedFillProperty

```
public static readonly BindableProperty SelectedFillProperty
```

Field Value

[BindableProperty](#)

## SelectedStrokeProperty

```
public static readonly BindableProperty SelectedStrokeProperty
```

Field Value

[BindableProperty](#)

## SelectedStrokeWidthProperty

```
public static readonly BindableProperty SelectedStrokeWidthProperty
```

Field Value

[BindableProperty↗](#)

## UnSelectedFillProperty

```
public static readonly BindableProperty UnSelectedFillProperty
```

Field Value

[BindableProperty↗](#)

## UnSelectedStrokeProperty

```
public static readonly BindableProperty UnSelectedStrokeProperty
```

Field Value

[BindableProperty↗](#)

## UnSelectedStrokeWidthProperty

```
public static readonly BindableProperty UnSelectedStrokeWidthProperty
```

Field Value

[BindableProperty↗](#)

## ValueProperty

```
public static readonly BindableProperty ValueProperty
```

Field Value

[BindableProperty](#)

## Properties

Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

IsReadOnly

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

Property Value

[bool](#)

ItemCount

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

Property Value

[int](#)

RatingDrawable

```
public RatingDrawable RatingDrawable { get; set; }
```

Property Value

[RatingDrawable](#)

## SelectedFill

```
public Color SelectedFill { get; set; }
```

Property Value

[Color](#)

## SelectedStroke

```
public Color SelectedStroke { get; set; }
```

Property Value

[Color](#)

## SelectedStrokeWidth

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

Property Value

[double](#)

## UnSelectedFill

```
public Color UnSelectedFill { get; set; }
```

Property Value

[Color](#)

## UnSelectedStroke

```
public Color UnSelectedStroke { get; set; }
```

Property Value

[Color](#)

## UnSelectedStrokeWidth

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

Property Value

[double](#)

## Value

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

Property Value

[int](#)

## Methods

### OnParentSet()

Raises the (internal) **ParentSet** event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet\(\)](#)

## Events

### ValueChanged

```
public event EventHandler<RatingValueChangedEventArgs> ValueChanged
```

#### Event Type

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

# Class RatingDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class RatingDrawable : IDrawable
```

## Inheritance

[object](#) ← RatingDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## ItemCount

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

### Property Value

[int](#)

## SelectedFillColor

```
public Color SelectedFillColor { get; set; }
```

Property Value

[Color](#) ↗

## SelectedStrokeColor

```
public Color SelectedStrokeColor { get; set; }
```

Property Value

[Color](#) ↗

## SelectedStrokeWidth

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

Property Value

[double](#) ↗

## UnSelectedFillColor

```
public Color UnSelectedFillColor { get; set; }
```

Property Value

[Color](#) ↗

## UnSelectedStrokeColor

```
public Color UnSelectedStrokeColor { get; set; }
```

Property Value

[Color](#)

## UnSelectedStrokeWidth

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

Property Value

[double](#)

## Value

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

Property Value

[int](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawRatingItem(ICanvas, RectF, int)

```
public virtual void DrawRatingItem(ICanvas canvas, RectF dirtyRect, int index)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

index [int](#)

# Class RatingValueChangedEventArgs

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The RatingValueChangedEventArgs class provides data for events that report a change in the rating value of a Rating control.

```
public class RatingValueChangedEventArgs : EventArgs
```

## Inheritance

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

## Inherited Members

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

## Constructors

### RatingValueChangedEventArgs(double)

```
public RatingValueChangedEventArgs(double value)
```

## Parameters

**value** [double](#)

## Properties

### Value

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

### Property Value

double ↗

# Class SegmentedControl

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The SegmentedControl is a drawn control that allows users to make a single selection from a set of segmented options. It is ideal for scenarios where a user needs to choose between mutually exclusive options, such as switching between tabs or filtering content.

```
public class SegmentedControl : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← SegmentedControl
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate\(\) ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState\(\) ↩ , View.GetChildElements\(Point\) ↩ , View.OnBindingContextChanged\(\) ↩ ,
View.GetRendererOverrides<T>\(\) ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
```

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.SizeAllocated\(double, double\)](#) ,  
[VisualElement.RefreshIsEnabledProperty\(\)](#) , [VisualElement.Arrange\(Rect\)](#) ,  
[VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### SegmentedControl()

```
public SegmentedException()
```

## Fields

### ActiveBackgroundProperty

```
public static readonly BindableProperty ActiveBackgroundProperty
```

## Field Value

[BindableProperty](#)

## ActiveFontSizeProperty

```
public static readonly BindableProperty ActiveFontSizeProperty
```

Field Value

[BindableProperty](#)

## ActiveTextColorProperty

```
public static readonly BindableProperty ActiveTextColorProperty
```

Field Value

[BindableProperty](#)

## BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

Field Value

[BindableProperty](#)

## CornerRadiusProperty

```
public static readonly BindableProperty CornerRadiusProperty
```

Field Value

[BindableProperty](#)

## FontSizeProperty

```
public static readonly BindableProperty FontSizeProperty
```

Field Value

[BindableProperty↗](#)

## ItemsSourceProperty

```
public static readonly BindableProperty ItemsSourceProperty
```

Field Value

[BindableProperty↗](#)

## SelectedIndexChangedCommandProperty

```
public static readonly BindableProperty SelectedIndexChangedCommandProperty
```

Field Value

[BindableProperty↗](#)

## SelectedIndexProperty

```
public static readonly BindableProperty SelectedIndexProperty
```

Field Value

[BindableProperty↗](#)

## TextColorProperty

```
public static readonly BindableProperty TextColorProperty
```

Field Value

[BindableProperty](#)

## Properties

ActiveBackground

```
public Brush ActiveBackground { get; set; }
```

Property Value

[Brush](#)

ActiveFontSize

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

Property Value

[double](#)

ActiveTextColor

```
public Color ActiveTextColor { get; set; }
```

Property Value

[Color](#)

Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

## CornerRadius

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

Property Value

[double](#)

## FontSize

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

Property Value

[double](#)

## ItemsSource

```
public IEnumerable ItemsSource { get; set; }
```

Property Value

[IEnumerable](#)

## SegmentedControlDrawable

```
public SegmentedControlDrawable SegmentedControlDrawable { get; set; }
```

Property Value

[SegmentedControlDrawable](#)

## SelectedIndex

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

Property Value

[int](#)

## SelectedIndexChangedCommand

```
public ICommand SelectedIndexChangedCommand { get; set; }
```

Property Value

[ICommand](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#)

## Methods

### OnParentSet()

Raises the (internal) `ParentSet` event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet\(\)](#)

## OnSizeAllocated(double, double)

Method that is called when the size of the element is set during a layout cycle. Implement this method to add class handling for this event.

```
protected override void OnSizeAllocated(double width, double height)
```

## Parameters

`width` [double](#)

The new width of the element.

`height` [double](#)

The new height of the element.

## Events

### SelectedIndexChanged

```
public event EventHandler<SelectedIndexEventArgs> SelectedIndexChanged
```

## Event Type

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

# Class SegmentedControlDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class SegmentedControlDrawable : IDrawable
```

## Inheritance

[object](#) ← SegmentedControlDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

## Properties

### ActiveBackgroundPaint

```
public Paint ActiveBackgroundPaint { get; set; }
```

#### Property Value

[Paint](#)

### ActiveFontSize

```
public float ActiveFontSize { get; set; }
```

#### Property Value

[float](#)

## ActiveTextColor

```
public Color ActiveTextColor { get; set; }
```

Property Value

[Color](#)

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

Property Value

[Paint](#)

## CornerRadius

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

Property Value

[double](#)

## FontSize

```
public float FontSize { get; set; }
```

Property Value

[float](#)

## ItemsSource

```
public IEnumerable ItemsSource { get; set; }
```

Property Value

[IEnumerable](#)

## SelectedIndex

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

Property Value

[int](#)

## TextColor

```
public Color TextColor { get; set; }
```

Property Value

[Color](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawActiveTab(ICanvas, RectF)

```
public virtual void DrawActiveTab(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawTabs(ICanvas, RectF)

```
public virtual void DrawTabs(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Class SelectedIndexEventArgs

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The SelectedIndexEventArgs class provides data for events that report a change in the selected index of a SegmentedControl.

```
public class SelectedIndexEventArgs : EventArgs
```

## Inheritance

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

## Inherited Members

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

## Constructors

### SelectedIndexEventArgs(int)

```
public SelectedIndexEventArgs(int selectedIndex)
```

## Parameters

selectedIndex [int](#)

## Properties

### SelectedIndex

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

## Property Value

int ↗

# Class Slider

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The Slider class is a graphical control that allows users to select a value from a continuous or discrete range by dragging a thumb along a track. This versatile control is widely used for adjusting settings like volume, brightness, or custom application parameters.

```
public class Slider : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← Slider

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### Slider()

```
public Slider()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

### Field Value

[BindableProperty](#)

## MaximumBrushProperty

```
public static readonly BindableProperty MaximumBrushProperty
```

Field Value

[BindableProperty](#)

## MaximumProperty

```
public static readonly BindableProperty MaximumProperty
```

Field Value

[BindableProperty](#)

## MinimumBrushProperty

```
public static readonly BindableProperty MinimumBrushProperty
```

Field Value

[BindableProperty](#)

## MinimumProperty

```
public static readonly BindableProperty MinimumProperty
```

Field Value

[BindableProperty](#)

## ThumbBrushProperty

```
public static readonly BindableProperty ThumbBrushProperty
```

Field Value

[BindableProperty↗](#)

## ThumbShapeProperty

```
public static readonly BindableProperty ThumbShapeProperty
```

Field Value

[BindableProperty↗](#)

## ValueProperty

```
public static readonly BindableProperty ValueProperty
```

Field Value

[BindableProperty↗](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush↗](#)

## Maximum

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

Property Value

[double](#) ↗

## MaximumBrush

```
public Brush MaximumBrush { get; set; }
```

Property Value

[Brush](#) ↗

## Minimum

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

Property Value

[double](#) ↗

## MinimumBrush

```
public Brush MinimumBrush { get; set; }
```

Property Value

[Brush](#) ↗

## SliderDrawable

```
public SliderDrawable SliderDrawable { get; set; }
```

Property Value

[SliderDrawable](#)

## ThumbBrush

```
public Brush ThumbBrush { get; set; }
```

Property Value

[Brush](#)

## ThumbShape

```
public ThumbShape ThumbShape { get; set; }
```

Property Value

[ThumbShape](#)

## Value

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

Property Value

[double](#)

## Methods

### OnParentSet()

Raises the (internal) **ParentSet** event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet\(\)](#)

## Events

### ValueChanged

```
public event EventHandler<ValueChangedEventArgs> ValueChanged
```

#### Event Type

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

# Class SliderDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class SliderDrawable : IDrawable
```

## Inheritance

[object](#) ← SliderDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## Maximum

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

### Property Value

[double](#)

## MaximumPaint

```
public Paint MaximumPaint { get; set; }
```

Property Value

[Paint](#)

## Minimum

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

Property Value

[double](#)

## MinimumPaint

```
public Paint MinimumPaint { get; set; }
```

Property Value

[Paint](#)

## ThumbPaint

```
public Paint ThumbPaint { get; set; }
```

Property Value

[Paint](#)

## ThumbShape

```
public ThumbShape ThumbShape { get; set; }
```

Property Value

[ThumbShape](#)

Value

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

Property Value

[double](#)

## Methods

Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawThumb(ICanvas, RectF)

```
public virtual void DrawThumb(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawTrackBackground(ICanvas, RectF)

```
public virtual void DrawTrackBackground(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawTrackProgress(ICanvas, RectF)

```
public virtual void DrawTrackProgress(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Enum ThumbShape

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ThumbShape enumeration defines the possible shapes for a "thumb" in UI components, such as sliders, progress indicators, or drag handles. It provides developers with options to customize the appearance of the thumb based on design requirements.

```
public enum ThumbShape
```

## Fields

Circle = 0

Rectangle = 1

RoundedRectangle = 2

# Class ToggleSwitch

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

The ToggleSwitch is a drawn control that allows users to toggle between two states (on and off). Designed for visual clarity and smooth interactivity, this control is ideal for enabling or disabling settings, features, or options within an application.

```
public class ToggleSwitch : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

```
object ↩ ← BindableObject ↩ ← Element ↩ ← NavigableElement ↩ ← VisualElement ↩ ← View ↩ ←
GraphicsView ↩ ← ToggleSwitch
```

## Implements

```
INotifyPropertyChanged ↩ , IVisualTreeElement ↩ , IEffectControlProvider ↩ , IToolTipElement ↩ ,
IContextFlyoutElement ↩ , IAnimatable ↩ , IViewController ↩ , IVisualElementController ↩ ,
IElementController ↩ , IGestureController ↩ , IGestureRecognizers ↩ , IPropertyMapperView ↩ ,
IHotReloadableView ↩ , IReplaceableView ↩ , IGraphicsView ↩ , IView ↩ , IElement ↩ , ITransform ↩
```

## Inherited Members

```
GraphicsView.DrawableProperty ↩ , GraphicsView.Invalidate() ↩ , GraphicsView.Drawable ↩ ,
GraphicsView.StartHoverInteraction ↩ , GraphicsView.MoveHoverInteraction ↩ ,
GraphicsView.EndHoverInteraction ↩ , GraphicsView.StartInteraction ↩ , GraphicsView.DragInteraction ↩ ,
GraphicsView.EndInteraction ↩ , GraphicsView.CancelInteraction ↩ , View.VerticalOptionsProperty ↩ ,
View.HorizontalOptionsProperty ↩ , View.MarginProperty ↩ , View.propertyMapper ↩ ,
View.ChangeVisualState() ↩ , View.GetChildElements(Point) ↩ , View.OnBindingContextChanged() ↩ ,
View.GetRendererOverrides<T>() ↩ , View.GestureController ↩ , View.GestureRecognizers ↩ ,
View.HorizontalOptions ↩ , View.Margin ↩ , View.VerticalOptions ↩ , VisualElement.NavigationProperty ↩ ,
VisualElement.StyleProperty ↩ , VisualElement.InputTransparentProperty ↩ ,
VisualElement.IsEnabledProperty ↩ , VisualElement.XProperty ↩ , VisualElement.YProperty ↩ ,
VisualElement.AnchorXProperty ↩ , VisualElement.AnchorYProperty ↩ ,
VisualElement.TranslationXProperty ↩ , VisualElement.TranslationYProperty ↩ ,
VisualElement.WidthProperty ↩ , VisualElement.HeightProperty ↩ , VisualElement.RotationProperty ↩ ,
VisualElement.RotationXProperty ↩ , VisualElement.RotationYProperty ↩ , VisualElement.ScaleProperty ↩ ,
```

[VisualElement.ScaleXProperty](#) , [VisualElement.ScaleYProperty](#) , [VisualElement.ClipProperty](#) ,  
[VisualElement.VisualProperty](#) , [VisualElement.IsVisibleProperty](#) , [VisualElement.OpacityProperty](#) ,  
[VisualElement.BackgroundColorProperty](#) , [VisualElement.BehaviorsProperty](#) ,  
[VisualElement.TriggersProperty](#) , [VisualElement.WidthRequestProperty](#) ,  
[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.IsFocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.IsFocused](#) , [VisualElement.IsVisible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,

[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,  
[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### ToggleSwitch()

```
public ToggleSwitch()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

[BindableProperty](#)

## HasShadowProperty

```
public static readonly BindableProperty HasShadowProperty
```

Field Value

[BindableProperty](#)

## IsOnProperty

```
public static readonly BindableProperty IsOnProperty
```

Field Value

[BindableProperty](#)

## ThumbBrushProperty

```
public static readonly BindableProperty ThumbBrushProperty
```

Field Value

[BindableProperty](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

## HasShadow

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

Property Value

[bool](#)

## IsOn

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

Property Value

[bool](#)

## ThumbBrush

```
public Brush ThumbBrush { get; set; }
```

Property Value

[Brush](#)

## ToggleSwitchDrawable

```
public ToggleSwitchDrawable ToggleSwitchDrawable { get; set; }
```

Property Value

[ToggleSwitchDrawable](#)

# Methods

## OnParentSet()

Raises the (internal) **ParentSet** event.

```
protected override void OnParentSet()
```

## Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

## See Also

[OnParentSet\(\)](#)

# Events

## Toggled

```
public event EventHandler<ToggledEventArgs> Toggled
```

## Event Type

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

# Class ToggleSwitchDrawable

Namespace: [AlohaKit.Controls](#)

Assembly: AlohaKit.dll

```
public class ToggleSwitchDrawable : IDrawable
```

## Inheritance

[object](#) ← ToggleSwitchDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## HasShadow

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

### Property Value

[bool](#)

## IsOn

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

### Property Value

[bool](#)

## ThumbPaint

```
public Paint ThumbPaint { get; set; }
```

### Property Value

[Paint](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public virtual void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

`dirtyRect` [RectF](#)

## DrawThumb(ICanvas, RectF)

```
public virtual void DrawThumb(ICanvas canvas, RectF dirtyRect)
```

### Parameters

`canvas` [ICanvas](#)

`dirtyRect` [RectF](#)

# Namespace AlohaKit.DataVisualization

## Classes

[PieChart](#)

[PieChartDrawable](#)

# Class PieChart

Namespace: [AlohaKit.DataVisualization](#)

Assembly: AlohaKit.DataVisualization.dll

```
public class PieChart : GraphicsView, INotifyPropertyChanged, IVisualTreeElement,
IEffectControlProvider, IToolTipElement, IContextFlyoutElement, IAnimatable,
IViewController, IVisualElementController, IElementController, IGestureController,
IGestureRecognizers, IPropertyMapperView, IHotReloadableView, IReplaceableView,
IGraphicsView, IView, IElement, ITransform
```

## Inheritance

[object](#) ← [BindableObject](#) ← [Element](#) ← [NavigableElement](#) ← [VisualElement](#) ← [View](#) ←  
[GraphicsView](#) ← PieChart

## Implements

[INotifyPropertyChanged](#), [IVisualTreeElement](#), [IEffectControlProvider](#), [IToolTipElement](#),  
[IContextFlyoutElement](#), [IAnimatable](#), [IViewController](#), [IVisualElementController](#),  
[IElementController](#), [IGestureController](#), [IGestureRecognizers](#), [IPropertyMapperView](#),  
[IHotReloadableView](#), [IReplaceableView](#), [IGraphicsView](#), [IView](#), [IElement](#), [ITransform](#)

## Inherited Members

[GraphicsView.DrawableProperty](#), [GraphicsView.Invalidate\(\)](#), [GraphicsView.Drawable](#),  
[GraphicsView.StartHoverInteraction](#), [GraphicsView.MoveHoverInteraction](#),  
[GraphicsView.EndHoverInteraction](#), [GraphicsView.StartInteraction](#), [GraphicsView.DragInteraction](#),  
[GraphicsView.EndInteraction](#), [GraphicsView.CancelInteraction](#), [View.VerticalOptionsProperty](#),  
[View.HorizontalOptionsProperty](#), [View.MarginProperty](#), [View.propertyMapper](#),  
[View.ChangeVisualState\(\)](#), [View.GetChildElements\(Point\)](#), [View.OnBindingContextChanged\(\)](#),  
[View.GetRendererOverrides<T>\(\)](#), [View.GestureController](#), [View.GestureRecognizers](#),  
[View.HorizontalOptions](#), [View.Margin](#), [View.VerticalOptions](#), [VisualElement.NavigationProperty](#),  
[VisualElement.StyleProperty](#), [VisualElement.InputTransparentProperty](#),  
[VisualElement.IsEnabledProperty](#), [VisualElement.XProperty](#), [VisualElement.YProperty](#),  
[VisualElement.AnchorXProperty](#), [VisualElement.AnchorYProperty](#),  
[VisualElement.TranslationXProperty](#), [VisualElement.TranslationYProperty](#),  
[VisualElement.WidthProperty](#), [VisualElement.HeightProperty](#), [VisualElement.RotationProperty](#),  
[VisualElement.RotationXProperty](#), [VisualElement.RotationYProperty](#), [VisualElement.ScaleProperty](#),  
[VisualElement.ScaleXProperty](#), [VisualElement.ScaleYProperty](#), [VisualElement.ClipProperty](#),  
[VisualElement.VisualProperty](#), [VisualElement.IsVisibleProperty](#), [VisualElement.OpacityProperty](#),  
[VisualElement.BackgroundColorProperty](#), [VisualElement.BehaviorsProperty](#),  
[VisualElement.TriggersProperty](#), [VisualElement.WidthRequestProperty](#),

[VisualElement.HeightRequestProperty](#) , [VisualElement.MinimumWidthRequestProperty](#) ,  
[VisualElement.MinimumHeightRequestProperty](#) , [VisualElement.MaximumWidthRequestProperty](#) ,  
[VisualElement.MaximumHeightRequestProperty](#) , [VisualElement.FocusedProperty](#) ,  
[VisualElement.FlowDirectionProperty](#) , [VisualElement.WindowProperty](#) ,  
[VisualElement.ShadowProperty](#) , [VisualElement.ZIndexProperty](#) ,  
[VisualElement.ControlsVisualElementMapper](#) , [VisualElement.BatchBegin\(\)](#) ,  
[VisualElement.BatchCommit\(\)](#) , [VisualElement.Focus\(\)](#) ,  
[VisualElement.Measure\(double, double, MeasureFlags\)](#) , [VisualElement.Unfocus\(\)](#) ,  
[VisualElement.InvalidateMeasure\(\)](#) , [VisualElement.OnChildAdded\(Element\)](#) ,  
[VisualElement.OnChildRemoved\(Element, int\)](#) , [VisualElement.OnChildrenReordered\(\)](#) ,  
[VisualElement.OnMeasure\(double, double\)](#) , [VisualElement.OnSizeAllocated\(double, double\)](#) ,  
[VisualElement.SizeAllocated\(double, double\)](#) , [VisualElement.RefreshIsEnabledProperty\(\)](#) ,  
[VisualElement.Arrange\(Rect\)](#) , [VisualElement.ArrangeOverride\(Rect\)](#) , [VisualElement.Layout\(Rect\)](#) ,  
[VisualElement.InvalidateMeasureOverride\(\)](#) , [VisualElement.MeasureOverride\(double, double\)](#) ,  
[VisualElement.MapBackgroundColor\(IViewHandler, IView\)](#) ,  
[VisualElement.MapBackgroundImageSource\(IViewHandler, IView\)](#) , [VisualElement.Visual](#) ,  
[VisualElement.FlowDirection](#) , [VisualElement.Window](#) , [VisualElement.AnchorX](#) ,  
[VisualElement.AnchorY](#) , [VisualElement.BackgroundColor](#) , [VisualElement.Behaviors](#) ,  
[VisualElement.Bounds](#) , [VisualElement.Height](#) , [VisualElement.HeightRequest](#) ,  
[VisualElement.InputTransparent](#) , [VisualElement.IsEnabled](#) , [VisualElement.IsEnabledCore](#) ,  
[VisualElement.Focused](#) , [VisualElement.Visible](#) , [VisualElement.MinimumHeightRequest](#) ,  
[VisualElement.MinimumWidthRequest](#) , [VisualElement.MaximumHeightRequest](#) ,  
[VisualElement.MaximumWidthRequest](#) , [VisualElement.Opacity](#) , [VisualElement.Rotation](#) ,  
[VisualElement.RotationX](#) , [VisualElement.RotationY](#) , [VisualElement.Scale](#) , [VisualElement.ScaleX](#) ,  
[VisualElement.ScaleY](#) , [VisualElement.TranslationX](#) , [VisualElement.TranslationY](#) ,  
[VisualElement.Triggers](#) , [VisualElement.Width](#) , [VisualElement.WidthRequest](#) , [VisualElement.X](#) ,  
[VisualElement.Y](#) , [VisualElement.Clip](#) , [VisualElement.Resources](#) , [VisualElement.Frame](#) ,  
[VisualElement.Handler](#) , [VisualElement.Shadow](#) , [VisualElement.ZIndex](#) ,  
[VisualElement.DesiredSize](#) , [VisualElement.IsLoaded](#) , [VisualElement.ChildrenReordered](#) ,  
[VisualElement.Focused](#) , [VisualElement.MeasureInvalidated](#) , [VisualElement.SizeChanged](#) ,  
[VisualElement.Unfocused](#) , [VisualElement.Loaded](#) , [VisualElement.Unloaded](#) ,  
[NavigableElement.Navigation](#) , [NavigableElement.Style](#) , [NavigableElement.StyleClass](#) ,  
[NavigableElement.class](#) , [Element.AutomationIdProperty](#) , [Element.ClassIdProperty](#) ,  
[Element.ControlsElementMapper](#) , [Element.InsertLogicalChild\(int, Element\)](#) ,  
[Element.AddLogicalChild\(Element\)](#) , [Element.RemoveLogicalChild\(Element\)](#) ,  
[Element.ClearLogicalChildren\(\)](#) , [Element.FindByName\(string\)](#) ,  
[Element.RemoveDynamicResource\(BindableProperty\)](#) ,  
[Element.SetDynamicResource\(BindableProperty, string\)](#) , [Element.OnPropertyChanged\(string\)](#) ,  
[Element.OnParentChanging\(ParentChangingEventArgs\)](#) , [Element.OnParentChanged\(\)](#) ,  
[Element.OnHandlerChanging\(HandlerChangingEventArgs\)](#) , [Element.OnHandlerChanged\(\)](#) ,

[Element.MapAutomationPropertiesInAccessibleTree\(IElementHandler, Element\)](#) ,  
[Element.MapAutomationPropertiesExcludedWithChildren\(IElementHandler, Element\)](#) ,  
[Element.AutomationId](#) , [Element.ClassId](#) , [Element.Effects](#) , [Element.Id](#) , [Element.StyleId](#) ,  
[Element.Parent](#) , [Element.ChildAdded](#) , [Element.ChildRemoved](#) , [Element.DescendantAdded](#) ,  
[Element.DescendantRemoved](#) , [Element.ParentChanging](#) , [Element.ParentChanged](#) ,  
[Element.HandlerChanging](#) , [Element.HandlerChanged](#) , [BindableObject.BindingContextProperty](#) ,  
[BindableObject.ClearValue\(BindableProperty\)](#) , [BindableObject.ClearValue\(BindablePropertyKey\)](#) ,  
[BindableObject.GetValue\(BindableProperty\)](#) , [BindableObject.IsSet\(BindableProperty\)](#) ,  
[BindableObject.RemoveBinding\(BindableProperty\)](#) ,  
[BindableObject.SetBinding\(BindableProperty, BindingBase\)](#) , [BindableObject.ApplyBindings\(\)](#) ,  
[BindableObject.OnPropertyChanging\(string\)](#) , [BindableObject.UnapplyBindings\(\)](#) ,  
[BindableObject.SetValue\(BindableProperty, object\)](#) ,  
[BindableObject.SetValue\(BindablePropertyKey, object\)](#) ,  
[BindableObject.CoerceValue\(BindableProperty\)](#) , [BindableObject.CoerceValue\(BindablePropertyKey\)](#) ,  
[BindableObject.Dispatcher](#) , [BindableObject.BindingContext](#) , [BindableObject.PropertyChanged](#) ,  
[BindableObject.PropertyChanging](#) , [BindableObject.BindingContextChanged](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### PieChart()

```
public PieChart()
```

## Fields

### BackgroundProperty

```
public static readonly BindableProperty BackgroundProperty
```

## Field Value

[BindableProperty](#)

## ItemsSourceProperty

```
public static readonly BindableProperty ItemsSourceProperty
```

Field Value

[BindableProperty](#)

## ShowLabelsProperty

```
public static readonly BindableProperty ShowLabelsProperty
```

Field Value

[BindableProperty](#)

## Properties

### Background

```
public Brush Background { get; set; }
```

Property Value

[Brush](#)

### ItemsSource

```
public Dictionary<string, float> ItemsSource { get; set; }
```

Property Value

[Dictionary](#)<[string](#), [float](#)>

## PieChartDrawable

```
public PieChartDrawable PieChartDrawable { get; set; }
```

### Property Value

[PieChartDrawable](#)

## ShowLabels

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

### Property Value

[bool](#)

## Methods

### OnParentSet()

Raises the (internal) [ParentSet](#) event.

```
protected override void OnParentSet()
```

### Remarks

Will set the NavigationProxy's inner navigation object to closest topmost element capable of handling navigation calls.

### See Also

[OnParentSet\(\)](#)

# Class PieChartDrawable

Namespace: [AlohaKit.DataVisualization](#)

Assembly: AlohaKit.DataVisualization.dll

```
public class PieChartDrawable : IDrawable
```

## Inheritance

[object](#) ← PieChartDrawable

## Implements

[IDrawable](#)

## Inherited Members

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

# Properties

## BackgroundPaint

```
public Paint BackgroundPaint { get; set; }
```

### Property Value

[Paint](#)

## ItemsSource

```
public Dictionary<string, float> ItemsSource { get; set; }
```

### Property Value

[Dictionary](#)<[string](#), [float](#)>

## ShowLabels

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

### Property Value

[bool](#)

## Methods

### Draw(ICanvas, RectF)

```
public void Draw(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

[canvas](#) [ICanvas](#)

[dirtyRect](#) [RectF](#)

### DrawBackground(ICanvas, RectF)

```
public void DrawBackground(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

[canvas](#) [ICanvas](#)

[dirtyRect](#) [RectF](#)

### DrawLabels(ICanvas, RectF)

```
public void DrawLabels(ICanvas canvas, RectF dirtyRect)
```

#### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

## DrawSlices(ICanvas, RectF)

```
public void DrawSlices(ICanvas canvas, RectF dirtyRect)
```

### Parameters

canvas [ICanvas](#)

dirtyRect [RectF](#)

# Namespace AlohaKit.Enums

## Classes

[ChartEnums](#)

## Enums

[ChartEnums.LineChartStyle](#)

The LineChartStyle enumeration specifies the style used for rendering a line chart. It defines whether the chart should display straight lines or smooth curves to connect the data points.

# Class ChartEnums

Namespace: [AlohaKit.Enums](#)

Assembly: AlohaKit.dll

```
public class ChartEnums
```

## Inheritance

[object](#) ← ChartEnums

## Inherited Members

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

# Enum ChartEnums.LineChartStyle

Namespace: [AlohaKit.Enums](#)

Assembly: AlohaKit.dll

The LineChartStyle enumeration specifies the style used for rendering a line chart. It defines whether the chart should display straight lines or smooth curves to connect the data points.

```
public enum ChartEnums.LineChartStyle
```

## Fields

Curve = 1

Line = 0

# Namespace AlohaKit.Extensions

## Classes

[ChartExtensions](#)

[IEnumerableExtensions](#)

[NumericExtensions](#)

# Class ChartExtensions

Namespace: [AlohaKit.Extensions](#)

Assembly: AlohaKit.dll

```
public static class ChartExtensions
```

## Inheritance

[object](#) ← ChartExtensions

## Inherited Members

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

## Methods

### GetMaxValueForPositiveAxis(float)

Calculate the maximum value for horizontal or vertical lines for charts with Positive axis

```
public static float GetMaxValueForPositiveAxis(this float maxValue)
```

#### Parameters

**maxValue** [float](#)

point maximum from values

#### Returns

[float](#)

get the maximum value for the last line

### GetRangeValuesPositiveAndNegative(float, float)

Calculate the maximum range value for horizontal or vertical lines for charts with Positive and negative axis

```
public static float GetRangeValuesPositiveAndNegative(this float minValue, float maxValue)
```

Parameters

`minValue` [float](#)

point minimum from values

`maxValue` [float](#)

point maximum from values

Returns

[float](#)

get the maximum range for the entire chart

## GetStepForAxisPositiveAndNegative(float, float)

Compute the space between lines

```
public static float GetStepForAxisPositiveAndNegative(this float minValue, float maxValue)
```

Parameters

`minValue` [float](#)

Minimum value in negative axis

`maxValue` [float](#)

Maximum value in positive axis

Returns

[float](#)

## GetStepForPositiveAxis(float)

Compute the space between lines

```
public static float GetStepForPositiveAxis(this float maxValue)
```

Parameters

`maxValue` [float](#)

point maximum from values

Returns

[float](#)

Get the soace between lines

## Remove0AfterPoint(string)

Cleans extra 0 from a string when needed. Example 40.0 is parsed to 40

```
public static string Remove0AfterPoint(this string number)
```

Parameters

`number` [string](#)

number string representation

Returns

[string](#)

String without extra 0 decimal

## ToKMBNumber(decimal, int, bool)

Formats a number to thousands (K),millions (M), billions (B) and trillions (T)

```
public static decimal ToKMBNumber(this decimal number, int decimalsToRound = 0, bool  
isPercentage = false)
```

## Parameters

number [decimal](#)

number to format

decimalsToRound [int](#)

number of decimal places to apply the rounding

isPercentage [bool](#)

the value is percentage

## Returns

[decimal](#)

Formatted KMB number

## ToKMBString(decimal, bool, bool)

Formats a number to thousands (K), millions (M) , billions (B) and trillions (T)

```
public static string ToKMBString(this decimal number, bool isRounded = false, bool  
isPercentage = false)
```

## Parameters

number [decimal](#)

string number representation to format

isRounded [bool](#)

check if it is needed the rounding

isPercentage [bool](#)

the value is percentage

Returns

[string ↗](#)

Formatted KMB number representation

# Class IEnumerableExtensions

Namespace: [AlohaKit.Extensions](#)

Assembly: AlohaKit.dll

```
public static class IEnumerableExtensions
```

## Inheritance

[object](#) ← IEnumerableExtensions

## Inherited Members

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

## Methods

### Count(IEnumerable)

```
public static int Count(this IEnumerable source)
```

#### Parameters

source [IEnumerable](#)

#### Returns

[int](#)

### ElementAt(IEnumerable, int)

```
public static object ElementAt(this IEnumerable source, int index)
```

#### Parameters

source [IEnumerable](#)

[index](#) [int ↗](#)

Returns

[object ↗](#)

# Class NumericExtensions

Namespace: [AlohaKit.Extensions](#)

Assembly: AlohaKit.dll

```
public static class NumericExtensions
```

## Inheritance

[object](#) ← NumericExtensions

## Inherited Members

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

# Methods

## Clamp(double, double, double)

```
public static double Clamp(this double self, double min, double max)
```

### Parameters

self [double](#)

min [double](#)

max [double](#)

### Returns

[double](#)

# Namespace AlohaKit.Models

## Classes

[ChartGroupStyle](#)

[ChartItem](#)

# Class ChartGroupStyle

Namespace: [AlohaKit.Models](#)

Assembly: AlohaKit.dll

```
public class ChartGroupStyle : INotifyPropertyChanged
```

## Inheritance

[object](#) ← ChartGroupStyle

## Implements

[INotifyPropertyChanged](#)

## Inherited Members

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

# Properties

## Background

Brush to be used for group background. Used for MultiLineChart

```
public Brush Background { get; set; }
```

## Property Value

[Brush](#)

## BackgroundColor

Color to be used for each group series background. Used for MultiLineChart

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

## Property Value

[Color](#)

## Color

```
public Color Color { get; set; }
```

Property Value

[Color](#)

## Id

Style identifier. Depending on chart type, current style will be applied globally/individually to the group entries that matches this Id

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

Property Value

[int](#)

## Methods

### OnPropertyChanged(string)

```
protected virtual void OnPropertyChanged(string propertyName = null)
```

Parameters

propertyName [string](#)

## Events

### PropertyChanged

Occurs when a property value changes.

```
public event PropertyChangedEventHandler PropertyChanged
```

Event Type

[PropertyChangedEventHandler](#)

# Class ChartItem

Namespace: [AlohaKit.Models](#)

Assembly: AlohaKit.dll

```
public class ChartItem : INotifyPropertyChanged
```

## Inheritance

[object](#) ← ChartItem

## Implements

[INotifyPropertyChanged](#)

## Inherited Members

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

# Properties

## GroupId

Group identifier. This value will be used internally to group entries by id

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

## Property Value

[int](#)

## IsLabelBold

Use Bold system font for this entry label

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

## Property Value

[bool](#)

## IsValueBold

Use Bold system font for this entry value

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

Property Value

[bool](#)

## Label

Footer value associated to current value

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

Property Value

[string](#)

## StyleId

Depending on the chart type, this property will be used to map each entry individual style to matching GroupStyles.Id style. For example, MultiBarChart will use this to set each bar colors

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

Property Value

[int](#)

## Value

Value to display

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

Property Value

[float](#)

## Methods

OnPropertyChanged(string)

```
protected virtual void OnPropertyChanged(string propertyName = null)
```

Parameters

propertyName [string](#)

## Events

PropertyChanged

Occurs when a property value changes.

```
public event PropertyChangedEventHandler PropertyChanged
```

Event Type

[PropertyChangedEventHandler](#)