

# API Reference

[Button](#)  
[ContentView](#)  
[FlowLayout](#)  
[Frame](#)  
[Group](#)  
[Image](#)  
[InputField](#)  
[Label](#)  
[List](#)  
[Mask](#)  
[Panel](#)  
[Region](#)  
[Slider](#)  
[View](#)  
[ViewSwitcher](#)

## Button

Inherits from [Frame](#)

### Description

Clickable region with text. Can be set to toggle and can have content.

### Fields

Text	string	Button text.
Disabled	bool	Boolean indicating if button is disabled.
IsToggleButton	bool	Boolean indicating if button is a toggle-button that can be toggled on/off.
ToggleValue	bool	Boolean indicating if button is toggled on (True) or off (False).
CanToggleOn	bool	Boolean indicating if button can be toggled on by the user (used if IsToggleButton=True).
CanToggleOff	bool	Boolean indicating if button can be toggled off by the user (used if IsToggleButton=True).
Font	Font	Button text font.
FontStyle	FontStyle	Button text font-style.
FontSize	int	Button text font-size.
FontColor	Color	Button text font-color.
TextAlignment	Alignment	Button text alignment.
ShadowColor	Color	Button text shadow color.
ShadowDistance	Vector2	Button text shadow distance.
OutlineColor	Color	Button text outline color.
OutlineDistance	Vector2	Button text outline distance.
HighlightedColor	Color	Highlighted button color.
PressedColor	Color	Pressed button color.
DisabledColor	Color	Disabled button color.
HighlightedImage	Sprite	Highlighted button image.
PressedImage	Sprite	Pressed button image.
DisabledImage	Sprite	Disabled button image.

### Actions

Click	Triggered when user clicks on the button.
MouseEnter	Triggered when pointer enters the button.
MouseExit	Triggered when the mouse exits the button.
MouseDown	Triggered on mouse down.
MouseUp	Triggered on mouse up.

### XML

Embedded XML

```
<Button Width="4em" Height="1em" Click="ButtonMouseClick" MouseEnter="ButtonMouseEnter"
        MouseExit="ButtonMouseExit" MouseDown="ButtonMouseDown" MouseUp="ButtonMouseUp">
  <Label Text="(text)" Font="{font}" FontStyle="{fontStyle}" FontSize="{fontSize}"
        FontColor="{fontColor}" TextAlignment="{textAlignment}" Width="100%" Height="100%"
        ShadowColor="{shadowColor}" ShadowDistance="{shadowDistance}"
        OutlineColor="{outlineColor}" OutlineDistance="{outlineDistance}" />
  <ContentContainer />
  <ViewAnimation id="highlightImageAnimation">
    <Animate Field="BackgroundImage" To="{highlightedImage}" />
  </ViewAnimation>
  <ViewAnimation id="pressedImageAnimation">
    <Animate Field="BackgroundImage" To="{pressedImage}" />
  </ViewAnimation>
  <ViewAnimation id="highlightColorAnimation">
    <Animate Field="BackgroundColor" To="{highlightedColor}" ReverseSpeed="0.5" Duration="0.85" />
  </ViewAnimation>
  <ViewAnimation id="pressedColorAnimation">
    <Animate Field="BackgroundColor" To="{pressedColor}" />
  </ViewAnimation>
</Button>
```

## ContentView

Inherits from [View](#)

### Description

View that allows content to be set. Has no additional logic of its own.

## FlowLayout

Inherits from [ContentView](#)

### Description

A content view that arranges its content into items that flow in a direction (horizontally or vertically). Items that doesn't fit on a row wraps around to the next row. It also provides logic for selecting items.

### Fields

SelectedIndex	string	Comma-separated list of indexes of items that should be selected by default.
CanDeselect	bool	Boolean indicating if items can be deselected.
CanMultiSelect	bool	Boolean indicating if multiple items can be selected.
CanSelect	bool	Boolean indicating if user can select items.
Items	Generic List	List of items to be arranged by the FlowList.
ItemColor	Color	Item color.
ItemImage	Sprite	Item image.
ItemImageType	Image.Type	Item image type.
SelectedItemColor	Color	Selected item color.
SelectedItemImage	Sprite	Selected item image.
HighlightedItemColor	Color	Highlighted item color.
HighlightedItemImage	Sprite	Highlighted item image.
ContentMargin	Margin	Margin of content.
Orientation	Orientation	Flow-direction of items.
Spacing	ElementSize	Spacing between items. Overrides horizontal/vertical spacing if set.
HorizontalSpacing	ElementSize	Horizontal spacing between items.
VerticalSpacing	ElementSize	Vertical spacing between items.

### Actions

SelectionChanged	Triggered when item-selection changes.
------------------	--

### Methods

```
/// <summary>
/// Selects item in list.
/// </summary>
/// <param name="objSource">Item to be selected.</param>
/// <param name="userTriggered">Boolean indicating if user triggered this action.</param>
public void SelectItem(object objSource, bool userTriggered = false);
```

### XML

Embedded XML

```
<FlowList Orientation="Horizontal">
  </FlowList>
```

## Frame

Inherits from [ContentView](#)

### Description

A content view that resizes itself to its content. Used to frame content.

### Fields

ResizeToContent	bool	Boolean indicating if content view should resize to its content.
ContentMargin	Margin	Margin of content.

### XML

Embedded XML

```
<Frame ResizeToContent="True" ContentMargin="0">
  <ContentContainer id="Content" Height="{ContentHeight}" ResizeToContent="{ResizeToContent}" />
</Frame>
```

## Group

Inherits from [ContentView](#)

### Description

A content view that arranges its content vertically or horizontally with a set spacing between. The content can be aligned Top/Bottom (if arranged Horizontally) or Left/Right (if arranged Vertically).

### Fields

Orientation	Orientation	Orientation to arrange content.
Spacing	ElementSize	Spacing between items.
ContentAlignment	Alignment	Determines how content is to be aligned. Content can be aligned Top/Bottom (if arranged horizontally) or Left/Right (if arranged vertically).

### XML

Embedded XML

```
<Group Spacing="0" Orientation="Vertical" />
```

## Image

Inherits from [View](#)

### Description

Used to display images. Adds the shorthands for setting background image and color.

### Fields

Color	Color	Background color overlay or tint of the view. Color values can be specified by name or hexcode (#aarrggbb or #rrggbb).
Path	Sprite	The background image of the view. The value is the path to the sprite asset.
Type	Image.Type	The image type of the background image: Simple, Sliced, Tiled or Filled.

### XML

Embedded XML

```
<Image />
```

## InputField

Inherits from [ContentView](#)

### Description

Input field where the user can type text. Supports single-/multi-line input and different input types (names, passwords, etc.). It allows for content that will be shown when the text is empty.

### Fields

Text	string	Text in the input field.
TextMargin	Margin	Margin of text in the input field.
Font	Font	Text font.
FontStyle	FontStyle	Text font-style.
FontSize	int	Text font-size.
LineSpacing	float	Text line spacing.
FontColor	Color	Text font-color.
CharacterLimit	int	Character limit.
ContentType	InputField.ContentType	Type of input content.
LineType	InputField.LineType	SingleLine, MultiLineSubmit or MultiLineNewline.
CaretBlinkRate	float	Blink rate of caret. Default rate is 0.7.
TextSelectionColor	Color	Background color of selected text.
SetValueOnEndEdit	bool	Boolean indicating if value should be set upon end edit or immediately while being typed.

### Actions

EndEdit	Triggered when user stops editing text in the input field.
---------	--

ValueChanged Triggered when text input changes. If SetValueOnEndEdit is True it's triggered only when edit ends.

### XML

Embedded XML

```
<InputField BackgroundColor="white" TextMargin="0,9,9,9" FontStyle="Normal" FontSize="18"
            LineSpacing="1" FontColor="black" Width="8em" Height="1em" CaretBlinkRate="1.7"
            BackgroundColor="{BackgroundColor}" ContentType="{ContentType}" LineType="{LineType}" TextSelectionColor="{aaaaaa}" />
<ContentContainer id="InputFieldPlaceholder" />
<Label id="InputText" Margin="{TextMargin}" Text="{Text}" RichText="false" TextAlignment="Left"
        Width="100%" Height="100%" Font="{font}" FontStyle="{FontStyle}" FontSize="{FontSize}"
        LineSpacing="{LineSpacing}" FontColor="{fontColor}" />
</InputField>
```

## Label

Inherits from [View](#)

### Description

Displays text. Can adjust its size to text width or height.

### Fields

Text	string	Label text.
AdjustToText	AdjustToText	Adjustes view size to text according to value: None, Width or Height.
Font	Font	Text font.
FontStyle	FontStyle	Text font-style.
FontSize	int	Text font-size.
LineSpacing	float	Text line spacing.
RichText	bool	Boolean indicating if label has rich text content.
TextAlignment	Alignment	Alignmnet of text.
FontColor	Color	Text font-color.
ShadowColor	Color	Text shadow color.
ShadowDistance	Vector2	Text shadow distance.
OutlineColor	Color	Text outline color.
OutlineDistance	Vector2	Text outline distance.

### XML

Embedded XML

```
<Label FontStyle="Normal" FontSize="18" LineSpacing="1" RichText="True" FontColor="black"
        TextAlignment="Left" Width="3em" Height="1em" />
```

## List

Inherits from [Group](#)

### Description

Expands on the [Group](#) view and adds selection-logic for selecting content. It's similar to the [FlowLayout](#) but doesn't wrap content and the selection mode is enabled by default.

### Fields

SelectedIndex	string	Comma-separated list of indexes of items that should be selected by default.
CanDeselect	bool	Boolean indicating if items can be deselected.
CanMultiSelect	bool	Boolean indicating if multiple items can be selected.
CanSelect	bool	Boolean indicating if user can select items.
Items	Generic List	List of items to be arranged by the FlowList.
ItemColor	Color	Item color.
ItemImage	Sprite	Item image.
ItemImageType	Image.Type	Item image type.
SelectedItemColor	Color	Selected item color.
SelectedItemImage	Sprite	Selected item image.
HighlightedItemColor	Color	Highlighted item color.
HighlightedItemImage	Sprite	Highlighted item image.

### Actions

SelectionChanged	Triggered when item-selection changes.
------------------	--

### Methods

```
/// <summary>
/// Selects item in list.
/// </summary>
/// <param name="objSource">Item to be selected.</param>
/// <param name="userTriggered">Boolean indicating if user triggered this action.</param>
public void SelectItem(object objSource, bool userTriggered = false);
```

### XML

Embedded XML

```
<List Spacing="0" Orientation="Vertical" CanDeselect="False" />
</List>
```

## Mask

Inherits from [ContentView](#)

### Description

Content view that masks its content using its background image.

### XML

Embedded XML

```
<Mask BackgroundColor="#80ffffff" />
```

## Panel

Inherits from [ContentView](#)

### Description

Content view that provides functionality for scrolling content. Scrollable content can be aligned using the ContentAlignment field.

### Fields

ScrollVertical	bool	Boolean indicating if content can be scrolled vertically.
ScrollHorizontal	bool	Boolean indicating if content can be scrolled horizontally.
ContentAlignment	Alignment	Content alignment. Determines initial scroll position.
ContentMargin	Margin	Margin of scrollable content.
HorizontalScrollBarImage	Sprite	Background image of horizontal scroll-bar.
VerticalScrollBarImage	Sprite	Background image of vertical scroll-bar.
HorizontalScrollBarImageType	Sprite	Background image type of horizontal scroll-bar.
VerticalScrollBarImageType	Sprite	Background image type of vertical scroll-bar.
HorizontalScrollBarHandleImage	Sprite	Background image of horizontal scroll-bar handle.
VerticalScrollBarHandleImage	Sprite	Background image of vertical scroll-bar handle.
HorizontalScrollBarHandleImageType	Sprite	Background image type of horizontal scroll-bar handle.
VerticalScrollBarHandleImageType	Sprite	Background image type of vertical scroll-bar handle.
HorizontalScrollBarColor	Color	Background color of horizontal scroll-bar.
VerticalScrollBarColor	Color	Background color of vertical scroll-bar.
HorizontalScrollBarHandleColor	Color	Background color of horizontal scroll-bar handle.
VerticalScrollBarHandleColor	Color	Background color of vertical scroll-bar handle.
ShowHorizontalScrollBar	bool	Boolean indicating if horizontal scroll-bar should be shown.
ShowVerticalScrollBar	bool	Boolean indicating if vertical scroll-bar should be shown.

### XML

Embedded XML

```
<Panel ScrollHorizontal="True" ScrollVertical="True" ContentMargin="0">
  <Region Margin="{ContentMargin}">
    <Mask Margin="0,0,0,0">
      <ScrollRect id="ScrollRect" ScrollArea="ScrollVertical" ScrollHorizontal="{ScrollHorizontal}" />
      <ContentContainer ResizeToContent="True" Alignment="{ContentAlignment}" />
    </ScrollRect>
  </Region>
  <Scrollbar id="HorizontalScrollbar" BackgroundImage="{HorizontalScrollbarImage}"
            BackgroundImageType="{HorizontalScrollbarImageType}"
            BackgroundColor="{HorizontalScrollbarColor}"
            ScrollBarHandleImage="{HorizontalScrollbarHandleImage}"
            ScrollBarHandleImageType="{HorizontalScrollbarHandleImageType}"
            BackgroundColor="{HorizontalScrollbarHandleColor}" />
  <Scrollbar id="VerticalScrollbar" Margin="0,0,0,0">
    <Region id="VerticalScrollbarRegion" BackgroundImage="{VerticalScrollbarImage}"
          BackgroundColor="{VerticalScrollbarColor}"
          ScrollBarHandleImage="{VerticalScrollbarHandleImage}"
          ScrollBarHandleImageType="{VerticalScrollbarHandleImageType}"
          ScrollBarHandleColor="{VerticalScrollbarHandleColor}" Orientation="Vertical"
          Margin="0,0,0,0" />
  </Region>
</Panel>
```

## Region

Inherits from [ContentView](#)

### Description

Content view used to spacially arrange views.

### XML

Embedded XML

```
<Region />
```

## Slider

Inherits from [View](#)

### Description

Slider with a handle that can be moved with the mouse. Can be oriented horizontally or vertically.

### Fields

SliderHandleLength	ElementSize	Length of slider handle.
SliderHandleBreadth	ElementSize	Breadth of slider handle.
SetValueOnDragEnded	bool	Boolean indicating if value is set on drag end or while dragging.
Length	ElementSize	Length of slider.
Breadth	ElementSize	Breadth of slider.
Orientation	Orientation	Orientation of slider.
SliderImage	Sprite	Background image of slider.
SliderColor	Color	Background color of slider.
SliderImageType	Image.Type	Background image type of slider.
SliderFillImage	Sprite	Background image of slider fill.
SliderFillColor	Color	Background color of slider fill.
SliderHandleImage	Sprite	Background image of slider handle.
SliderHandleColor	Color	Background color of slider handle.
SliderHandleImageType	Image.Type	Background image type of slider handle.
SliderFillMargin	Margin	Margin of fill image.
Min	float	Min value of slider.
Max	float	Max value of slider.
Value	float	Slider value.

### Actions

ValueChanged	Triggered when slider value changes. Triggered on end edit if SetValueOnEndEdit is True.
BeginDrag	Triggered when user begins to drag the slider.
EndDrag	Triggered when user ends to dragging the slider.
Drag	Triggered when user drags the slider.
InitializePotentialDrag	Triggered when user initializes a potential drag.

### XML

Embedded XML

```
<Slider Length="4em" Breadth="1em" Orientation="Horizontal" Drag="SliderDrag" BeginDrag="SliderBeginDrag"
        EndDrag="SliderEndDrag" InitializePotentialDrag="SliderInitializePotentialDrag"
        BackgroundColor="{BackgroundColor}"
        <Region id="SliderRegion" Margin="{SliderFillMargin}">
          <Image Path="{SliderImage}" Type="{SliderImageType}" Color="{SliderColor}" />
          <Image id="SliderFill" Path="{SliderFillImage}" Type="{SliderImageType}"
                Color="{SliderFillColor}" Alignment="Left" />
        </Region>
        <Image id="SliderHandle" Path="{SliderHandleImage}" Type="{SliderHandleImageType}"
              Color="{SliderHandleColor}" Width="{SliderHandleLength}"
              Height="{SliderHandleBreadth}" Alignment="Left" />
        </Region>
</Slider>
```

## View

### Description

Base class for all views.

### Fields

Width	ElementSize	Width of the view. Can be specified in pixels, percentage or elements. Default value: "100%".
Height	ElementSize	Height of the view. Can be specified in pixels, percentage or elements. Default value: "100%".
Alignment	Alignment	Alignment of the view: TopLeft, Top, TopRight, Left, Center, Right, BottomLeft, Bottom, BottomRight. Default value: "Center".
Margin	Margin	Specifies the view's margin from left, top, right and bottom. Default value: "0,0,0,0".
Offset	Margin	Specifies the view's offset from left, top, right and bottom. Default value: "0,0,0,0".
OffsetFromParent	Margin	Offset manipulated by parent views.
BackgroundColor	Color	Background color overlay or tint of the view. Color values can be specified by name or hexcode (#aarrggbb or #rrggbb). Default value: (not set).
BackgroundImage	Sprite	The background image of the view. The value is the path to the sprite asset. Default value: (not set).
BackgroundImageType	Image.Type	The image type of the background image: Simple, Sliced, Tiled or Filled. Default value: "Simple".
UpdateRectTransform	bool	Boolean indicating if RectTransform should be updated upon view update. True by default.
UpdateBackground	bool	Boolean indicating if background image/color should be updated upon view update. True by default.
Enabled	bool	Boolean indicating if view is enabled. Disabled views are hidden but still receive updates. True by default.
Id	string	View identifier.
Style	string	Name of style. May be used by themes to do selective styling.
Alpha	float	Alpha color channel of view (0-1).
Rotation	Vector3	X,Y,Z rotation of the view (in degrees).
Scale	Vector3	X,Y,Z scale of the view.
Scale	Vector3	X,Y,Z scale of the view.
Pivot	Vector2	Pivot position.

### Methods

```
/// <summary>
/// Sets the value of a field utilizing the binding and change tracking system.
/// </summary>
/// <typeparam name="TField">Type inferred from the field expression.</typeparam>
/// <param name="field">Field expression.</param>
/// <param name="value">Value to set.</param>
public void SetValue(TField field, ExpressionFunc(TField) field, object value);

/// <summary>
/// Gets boolean indicating if a field value has changed since last frame.
/// </summary>
/// <typeparam name="TField">Type inferred from the field expression.</typeparam>
/// <param name="field">Field expression.</param>
public bool HasChanged(TField field, ExpressionFunc(TField) field);

/// <summary>
/// Sets boolean indicating that a field value has changed since last frame.
/// </summary>
/// <typeparam name="TField">Type inferred from the field expression.</typeparam>
/// <param name="field">Field expression.</param>
public bool SetChanged(TField field, ExpressionFunc(TField) field);

/// <summary>
/// Updates layout and behavior of the view.
public void UpdateView();

/// <summary>
/// Updates the layout of the view and notifies parents of update.
public void UpdateLayouts();

/// <summary>
/// Updates the layout of the view.
public virtual void UpdateLayout();

/// <summary>
/// Updates the behavior of the view.
/// </summary>
public virtual void UpdateBehavior();

/// <summary>
/// Returns embedded XML for the view.
/// </summary>
public virtual void GetEmbeddedXML();
```

## ViewSwitcher

Inherits from [ContentView](#)

### Description

Content view that provides functionality for switching its content.

### Fields

StartView	string	Id of view to be displayed by default.
SwitchToDefault	bool	Boolean indicating if first view should be displayed by default.
TransitionIn	string	Id of view animation to apply on views transitioned to.
TransitionOut	string	Id of view animation to apply on views transitioned from.

### Methods

```
/// <summary>
/// Switches to a view.
/// </summary>
/// <param name="view">View to be switched to.</param>
/// <param name="animate">Boolean indicating if switch should be animated.</param>
public void SwitchToView(view, bool animate = true);

/// <summary>
/// Switches to a view.
/// </summary>
/// <param name="id">Id of view to be switched to.</param>
/// <param name="animate">Boolean indicating if switch should be animated.</param>
public void SwitchTo(string id, bool animate = true);

/// <summary>
/// Switches to a view.
/// </summary>
/// <param name="index">Zero-based index of view to be switched to.</param>
/// <param name="animate">Boolean indicating if switch should be animated.</param>
public void SwitchTo(int index, bool animate = true);
```

### XML

Embedded XML

```
<ViewSwitcher id="id" Default="true">
  <ContentContainer ResizeToContent="{false}" />
</ViewSwitcher>
```