

# ZirconSwitch

*Custom iOS-style switch control based on ZirconKit.*

## Introduction

The Xojo IDE uses an iOS replica switch in its inspector. It is the most requested custom control from the Xojo IDE. This ZirconSwitch is another version of the same control. It is not the same code from the Xojo IDE, it was created from the ground up based on ZirconKit, and has more features than the one in the Xojo IDE.

The ZirconSwitch can have custom captions and colors for both On and Off positions. You can set the width, or allow the control to size itself. The control is animated - which you can turn off - and supports Retina displays.

## The ZirconSwitch Class

Simply import the ZirconKit module, then the ZirconSwitch control and you're ready to go. All images are stored inside the ZirconSwitch class. Drag the control to your Window or ContainerControl, and start coding.

### Constants

#### AlignCenter

For use with the Align property, tells the switch to align to the center of the canvas.

#### AlignLeft

Aligns left.

#### AlignRight

Aligns right.

#### Revision

Returns the revision / version number of the control.

#### SizeMini

Used with the ControlSize property, sets the control height to 12 pixels.

#### SizeSmall

Used with the ControlSize property, sets the control height to 16 pixels.

### SizeNormal

Used with the ControlSize property, sets the control height to 20 pixels.

### StateOff

Used with the State property, sets the control to the false position.

### StateOn

Used with the State property, sets the control to the true position.

### StateIndeterminate

Used with the State property, sets the control to an indeterminate position. The thumb is in the middle of both positions.

## Events

### Action

The value of the switch has changed, either from user interaction, or from code.

### Paint (G As Graphics)

Allows drawing to the background of the control. The control will be drawn on top of whatever is drawn to G. Useful for ensuring the background / unused space matches your app on Windows, where transparency is not supported.

## Properties

### Align As Integer

Use the Align constants to specify how the control should draw itself inside the space.

### Animated As Boolean

Defaults to true, tells the control to animate value changes. While animation is enabled, value changes in code will trigger an animation. If setting the value from within an open event, good practice would be to disable animation, set the value, and enable animation again.

### ControlSize As Integer

Allows changing the height of the control. See the Size\* constants for possible values. Default is SizeNormal.

### ControlWidth As Integer

When set to -1, the control will size itself automatically based on the size of the captions. Otherwise, the value is the number of horizontal pixels the control should consume. Default value is -1.

### LeftSideCaption As String

The caption drawn on the left side of the switch thumb. This is the “true” caption. Default value is “On”.

#### LeftSideColor As Color

The background color of the left portion of the control.

#### RightSideCaption As String

The caption drawn on the right side of the switch thumb. This is the “false” caption. Default value is “Off”.

#### RightSideColor As Color

The background color of the right portion of the control.

#### State As Integer

Allows control over the state of the control, either Off, On, or Indeterminate.

When the value is On, the control is showing the left portion. Off shows the right portion. Indeterminate centers the thumb between both positions. Setting this value in code will trigger the animation if animation is enabled.

#### Value As Boolean

Convenience and compatibility property. Setting Value to true will set State to StateOn. Setting Value to false will set State to StateOff.