Class

UISegmentedControl

A `UISegmentedControl` object is a horizontal control made of multiple segments, each segment functioning as a discrete button. A segmented control affords a compact means to group together a number of controls.

Language

Swift | Objective-C

SDKs

iOS 2.0+

tvOS 2.0+

On This Page

Overview

Symbols

Relationships

See Also

Overview

A segmented control can display a title (an `NSString` object) or an image (`UIImage` object). The `UISegmentedControl` object automatically resizes segments to fit proportionally within their superview unless they have a specific width set. When you add and remove segments, you can request that the action be animated with sliding and fading effects.

You register the target-action methods for a segmented control using the `valueChanged` constant as shown below.

```
segmentedControl.addTarget(self, action: "action:", forControlEvents: .ValueChanged)
```

How you configure a segmented control can affect its display behavior:

- If you set a segmented control to have a momentary style, a segment doesn't show itself as selected (blue background) when the user touches it. The disclosure button is always momentary and doesn't affect the actual selection.

- In versions of iOS prior to 3.0, if a segmented control has only two segments, then it behaves like a switch—tapping the currently-selected segment causes the other segment to be selected. On iOS 3.0 and later, tapping the currently-selected segment does not cause the other segment to be selected.

Customizing Appearance

In iOS v5.0 and later, you can customize the appearance of segmented controls using the methods listed in Customizing Appearance. You can customize the appearance of all segmented controls using the appearance proxy (for example, `[UISegmentedControl appearance]`), or just of a single control.

When customizing appearance, in general, you should specify a value for the normal state of a property to be used by other states which don't have a custom value set. Similarly, when a property is dependent on the bar metrics (on the iPhone in landscape orientation, bars have a different height from standard), you should make sure you specify a value for `default`.

In the case of the segmented control, appearance properties for `landscapePhone` are only respected for segmented controls in the smaller navigation and toolbars that are used in landscape orientation on the iPhone.

To provide complete customization, you need to provide divider images for different state combinations, using `setDividerImage(_:forLeftSegmentState:rightSegmentState:barMetrics:)`:

```
// Image between two unselected segments.
mySegmentedControl.setDividerImage(myImage, forLeftSegmentState: UIControlState.Norm
                                   rightSegmentState: UIControlState.Normal, barMetr

// Image between segment selected on the left and unselected on the right.
mySegmentedControl.setDividerImage(myImage, forLeftSegmentState: UIControlState.Sele
                                   rightSegmentState: UIControlState.Normal, barMetr

// Image between segment selected on the right and unselected on the left.
mySegmentedControl.setDividerImage(myImage, forLeftSegmentState: UIControlState.Norm
                                   rightSegmentState: UIControlState.Selected, barMe
```

For more information about appearance and behavior configuration, see Segmented Controls.

Symbols

Initializing a Segmented Control

`init(items: [Any]?)`

Initializes and returns a segmented control with segments having the given titles or images.

Managing Segment Content

`func setImage(UIImage?, forSegmentAt: Int)`

Sets the content of a segment to a given image.

```
func imageForSegment(at: Int)
```
Returns the image for a specific segment

```
func setTitle(String?, forSegmentAt: Int)
```
Sets the title of a segment.

```
func titleForSegment(at: Int)
```
Returns the title of the specified segment.

Managing Segments

```
func insertSegment(with: UIImage?, at: Int, animated: Bool)
```
Inserts a segment at a specified position in the receiver and gives it an image as content.

```
func insertSegment(withTitle: String?, at: Int, animated: Bool)
```
Inserts a segment at a specific position in the receiver and gives it a title as content.

```
var numberOfSegments: Int
```
Returns the number of segments the receiver has.

```
func removeAllSegments()
```
Removes all segments of the receiver

```
func removeSegment(at: Int, animated: Bool)
```
Removes the specified segment from the receiver, optionally animating the transition.

```
var selectedSegmentIndex: Int
```
The index number identifying the selected segment (that is, the last segment touched).

Managing Segment Behavior and Appearance

```
var isMomentary: Bool
```
A Boolean value that determines whether segments in the receiver show selected state.

```
func setEnabled(Bool, forSegmentAt: Int)
```
Enables the specified segment.

```
func isEnabledForSegment(at: Int)
```
Returns whether the indicated segment is enabled.

```
func setContentOffset(CGSize, forSegmentAt: Int)
```
Adjusts the offset for drawing the content (image or text) of the specified segment.

```
func contentOffsetForSegment(at: Int)
```
Returns the offset for drawing the content (image or text) of the specified segment.

```
func setWidth(CGFloat, forSegmentAt: Int)
```
Sets the width of the specified segment of the receiver.

```
func widthForSegment(at: Int)
```
Returns the width of the indicated segment of the receiver.

```
var apportionsSegmentWidthsByContent: Bool
```
Indicates whether the control attempts to adjust segment widths based on their content widths.

Customizing Appearance

```
var tintColor: UIColor!
```
The tint color to apply to key elements in the segmented control.

```
func backgroundImage(for: UIControlState, barMetrics: UIBarMetrics)
```

Returns the background image for a given state and bar metrics.

```
func setBackgroundImage(UIImage?, for: UIControlState, barMetrics: UIB
arMetrics)
```

Sets the background image for a given state and bar metrics.

```
func contentPositionAdjustment(forSegmentType: UISegmentedControlSegme
nt, barMetrics: UIBarMetrics)
```

Returns the positioning offset for a given segment and bar metrics.

```
func setContentPositionAdjustment(UIOffset, forSegmentType: UISegmente
dControlSegment, barMetrics: UIBarMetrics)
```

Returns the content positioning offset for a given segment and bar metrics.

```
func dividerImage(forLeftSegmentState: UIControlState, rightSegmentSta
te: UIControlState, barMetrics: UIBarMetrics)
```

Returns the divider image used for a given combination of left and right segment states and bar metrics.

```
func setDividerImage(UIImage?, forLeftSegmentState: UIControlState, ri
ghtSegmentState: UIControlState, barMetrics: UIBarMetrics)
```

Sets the divider image used for a given combination of left and right segment states and bar metrics.

```
func titleTextAttributes(for: UIControlState)
```

Returns the text attributes of the title for a given control state.

```
func setTitleTextAttributes([AnyHashable : Any]?, for: UIControlState)
```

Sets the text attributes of the title for a given control state.

Constants

Segment Selection

A constant for indicating that no segment is selected.

`UISegmentedControlSegment`

Constants for specifying a segment in a control

Relationships

Inherits From

`UIControl`

Conforms To

`NSCoding`

See Also

Related Documentation

Segmented Controls