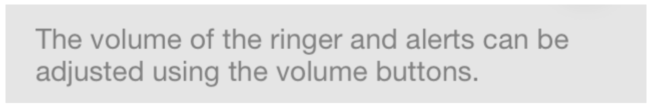
Labels

A label displays static text. Labels are often used in conjunction with controls to describe their intended purpose, such as explaining which value a button or slider affects.

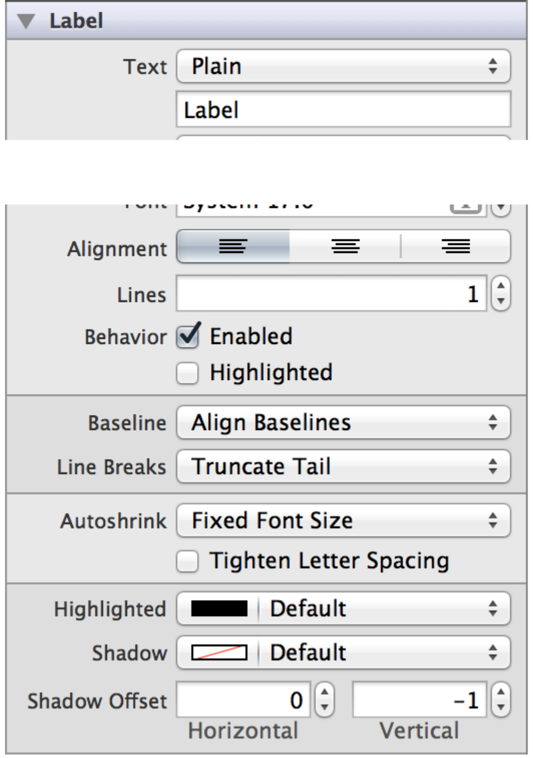

The volume of the ringer and alerts can be adjusted using the volume buttons.

Purpose. Labels allow the user to:

- Understand the purpose of controls in an app
- Receive instructions or context in an app

Implementation. Labels are implemented in the `UILabel` class and discussed in the *UILabel Class Reference*.

Configuration. Configure labels in Interface Builder, in the Label section of the Attributes Inspector. A few configurations cannot be made through the Attributes Inspector, so you must make them programmatically. You can set other configurations programmatically, too, if you prefer.

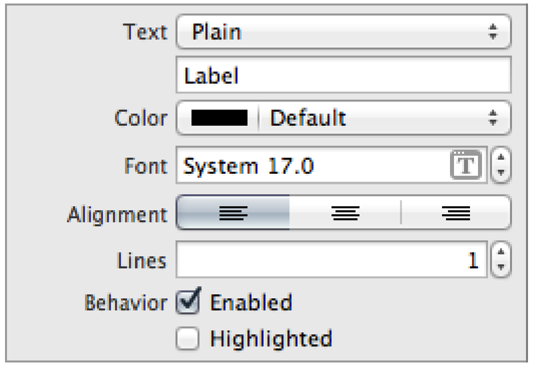

Label

Text	Plain
	Label
Alignment	left / center / right
Lines	1
Behavior	✓ Enabled
	☐ Highlighted
Baseline	Align Baselines
Line Breaks	Truncate Tail
Autoshrink	Fixed Font Size
	☐ Tighten Letter Spacing
Highlighted	■ Default
Shadow	▭ Default
Shadow Offset	0 Horizontal, -1 Vertical

On This Page

Content of Labels

Set label content using the Label Text (`text` and `attributedText`) field in the Attributes Inspector. Both properties get set whether you specified the value of the Text field to be plain or attributed. For more information about attributed text, see Specifying Text Appearance.

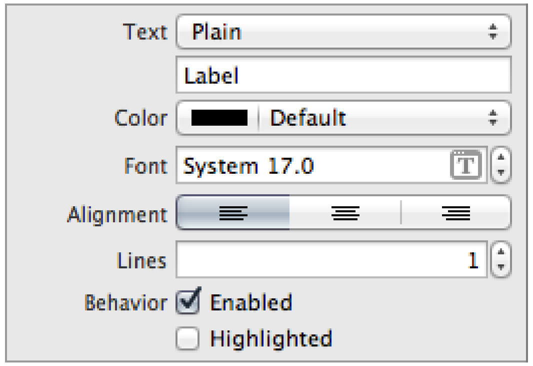


Text	Plain
	Label
Color	■ Default
Font	System 17.0
Alignment	left / center / right
Lines	1
Behavior	✓ Enabled
	☐ Highlighted

By default, a label is a single line. To create a multiline label, increase the value of the Lines (`numberOfLines`) field.

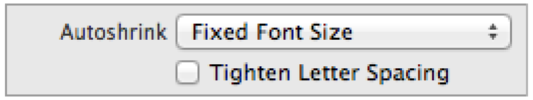
Behavior of Labels

You can specify whether a label is enabled or highlighted using the Enabled (`enabled`) and Highlighted (`highlighted`) checkboxes in the Attributes Inspector.



Text	Plain
	Label
Color	Default
Font	System 17.0
Alignment	
Lines	1
Behavior	✓ Enabled
	☐ Highlighted

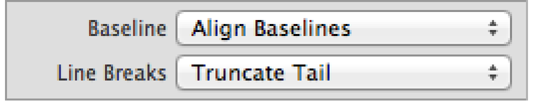
The Autoshrink (`adjustsFontSizeToFitWidth`) field is used to specify the manner in which font size will be reduced with the label's bounding rectangle.



Autoshrink	Fixed Font Size
	☐ Tighten Letter Spacing

The Fixed Font Size option is the equivalent of setting `adjustsFontSizeToFitWidth` to `NO`, meaning that font size will not adjust. Select the Minimum Font Scale (`minimumScaleFactor`) option to specify the smallest multiplier for the current font size that the font can scale down to, and the Minimum Font Size (`minimumFontSize`) option to specify the smallest font size that the font can scale down to.

Select the Tighten Letter Spacing (`adjustsLetterSpacingToFitWidth`) checkbox if you want the spacing between letters to be adjusted to fit the string within the label's bounding rectangle.

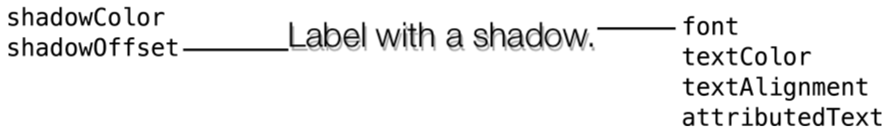

Baseline: Align Baselines
Line Breaks: Truncate Tail

The Baselines (baselineAdjustment) field determines how to adjust the position of text in cases when the text must be drawn using a different font size than the one originally specified. For example, with the Align Baselines option, the position of the baseline remains fixed at its initial location while the text appears to move toward that baseline. Similarly, selecting the None option makes it appear as if the text is moving upwards toward the top-left corner of the bounding box.

Use the Line Breaks (lineBreakMode) field to specify the technique to use for wrapping and truncating the label’s text if it exceeds a single line. Note that if this property is set to a value that causes text to wrap to another line, do not set the adjustsFontSizeToFitWidth or adjustsLetterSpacingToFitWidth property to YES.

Appearance of Labels

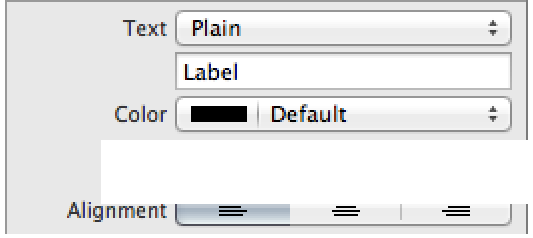
You can customize the appearance of a label by setting the properties depicted below.

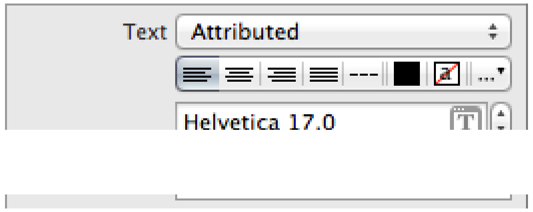


To customize the appearance of all labels in your app, use the appearance proxy (for example, [UILabel appearance]). For more information about appearance proxies, see Appearance Proxies.

Text Appearance

Labels can have one of two types of text: plain or attributed. Plain text supports a single set of formatting attributes—font, size, color, and so on—for the entire string. On the other hand, attributed text supports multiple sets of attributes that apply to individual characters or ranges of characters in the string.

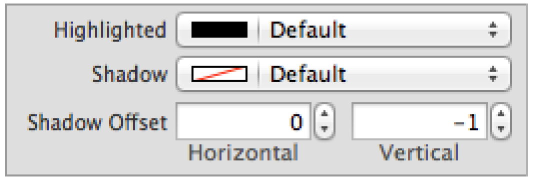

Text Plain
Label
Color Default
Alignment


Text Attributed
Helvetica 17.0

On This Page

Highlighted Appearance

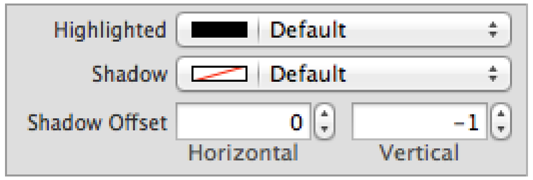
By default, the highlighted appearance of a label is no different than that of its normal control state.


Highlighted Default
Shadow Default
Shadow Offset 0 Horizontal -1 Vertical

However, you can create a different look for your label when it’s in the `UIControlStateHighlighted` state by modifying its color in the Highlighted (`highlightedTextColor`) field.

Text Shadow

You can set a color for your label’s shadow using the Shadow (`shadowColor`) field in the Attributes Inspector.


Highlighted Default
Shadow Default
Shadow Offset 0 Horizontal -1 Vertical

If you want to change how far the shadow is drawn from the button text, you can adjust the shadow offset. You can customize the offset for both dimensions using the Shadow Offset (`shadowOffset`) fields.

Using Auto Layout with Labels

On This Page

You can create [illegible]
any type of constraint for a label.

For general information about using Auto Layout with iOS views, see Using Auto Layout with Views.

Making Labels Accessible

Labels are accessible by default. The default accessibility trait for a label are Static Text and User Interaction Enabled.

For general information about making iOS views accessible, see Making Views Accessible.

Internationalizing Labels

For more information, see *Internationalization and Localization Guide*.

Debugging Labels

When debugging issues with labels, watch for this common pitfall:

Specifying conflicting text wrapping and font adjustment settings. The `lineBreakMode` property describes how text should wrap or truncate within the label. If you set a value for this property that causes text to wrap to another line, do not set the `adjustsFontSizeToFitWidth` and `adjustsLetterSpacingToFitWidth` properties to `YES`, those fields are used to scale the font size to fit into the label without adding line breaks.

Elements Similar to a Label

The following element provides similar functionality to a label:

Text Field. Text fields allows the user to input a single line of text into an app. You typically use text fields to gather small amounts of text from the user and perform some immediate action, such as a search operation, based on that te

On This Page