

Advanced Attributed Strings for iOS

Desktop class effects on a mobile device

Session 230

Ian Baird

Label Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Agenda

Agenda

- Advanced Attributes

Agenda

- Advanced Attributes
- Text effects in your App

Agenda

- Advanced Attributes
- Text effects in your App
- Contexts and Options

Advanced Attributes

Advanced Attributes

Common attributes

Advanced Attributes

Common attributes

Big red dog.

Advanced Attributes

Common attributes

NSFontAttributeName

Big red dog.

Advanced Attributes

Common attributes

NSFontAttributeName

NSForegroundColorAttributeName

Big red dog.

Advanced Attributes

Common attributes

NSFontAttributeName

NSForegroundColorAttributeName

NSParagraphStyleAttributeName

Big red dog.

Advanced Attributes

Common attributes

NSFontAttributeName

NSForegroundColorAttributeName

NSParagraphStyleAttributeName

alignment

Big red dog.

Advanced Attributes

Common attributes

NSFontAttributeName

NSForegroundColorAttributeName

NSParagraphStyleAttributeName

alignment

lineBreakMode

Big red
dog.

Advanced Attributes

`NSStrokeColorAttributeName` — `NSStrokeWidthAttributeName`



Advanced Attributes

`NSStrokeColorAttributeName` — `NSStrokeWidthAttributeName`



Advanced Attributes

NSStrokeWidthAttribute Name

Advanced Attributes

`NSStrokeWidthAttribute` Name

Positive width (stroke only) — @3.0



Advanced Attributes

NSStrokeWidthAttribute Name

Positive width (stroke only) — @3.0

Negative width (stroke & fill) — @-3.0



Advanced Attributes

NSStrokeWidthAttribute Name

Positive width (stroke only)—@3.0

Negative width (stroke & fill)—@-3.0

None—nil

A large, white, serif letter 'S' is centered on a black background. The letter is rendered with a thick stroke, demonstrating the effect of the NSStrokeWidthAttribute.

Advanced Attributes

`NSShadowAttributeName`



Advanced Attributes

NSShadowAttributeName



Advanced Attributes

NSShadow

- Value class in iOS

Shadow

Advanced Attributes

NSShadow

- Value class in iOS
shadowColor

Shadow

Advanced Attributes

NSShadow

- Value class in iOS

`shadowColor`
`shadowOffset`

Shadow

Advanced Attributes

NSShadow

- Value class in iOS

`shadowColor`

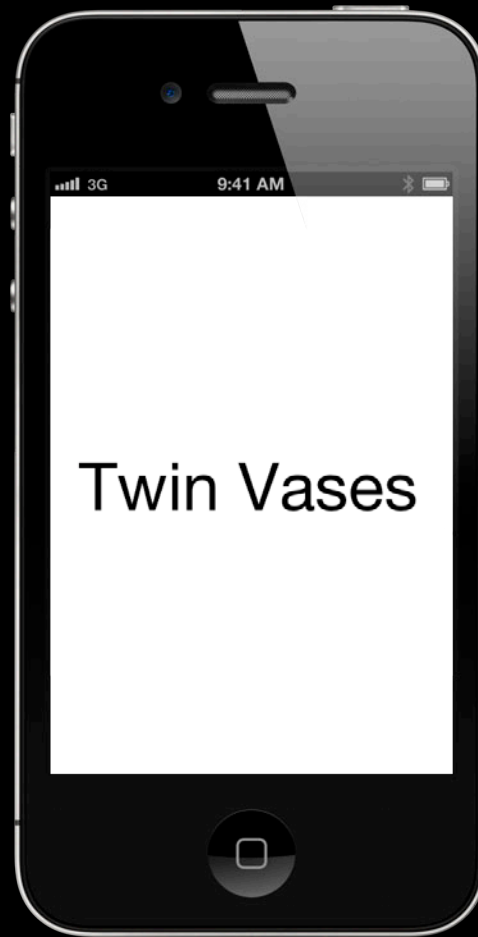
`shadowOffset`

`shadowBlurRadius`

Shadow

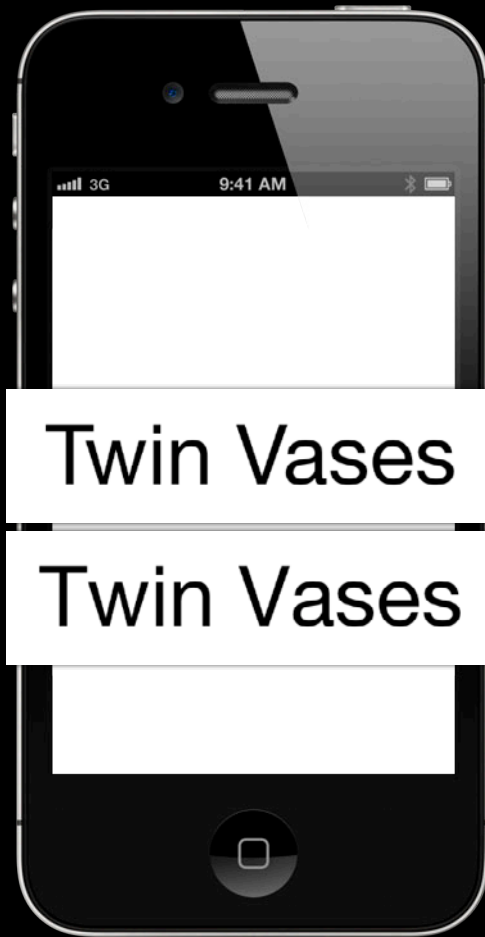
Advanced Attributes

NSKernAttributeName



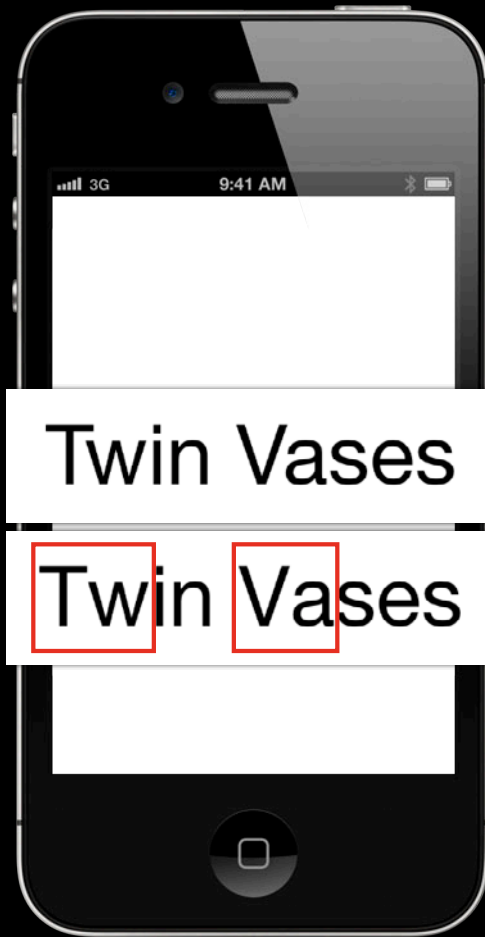
Advanced Attributes

NSKernAttributeName



Advanced Attributes

NSKernAttributeName



Advanced Attributes

NSKernAttributeName

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font
- Enable auto-kerning with `nil` or `[NSNull null]`

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font
- Enable auto-kerning with `nil` or `[NSNull null]`
- Disable kerning – `@0.0`

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font
- Enable auto-kerning with `nil` or `[NSNull null]`
- Disable kerning – `@0.0`
- Any other value adjusts the kerning by a number of points

```
@{NSKernAttributeName: [NSNull null]}
```

Big *red* dog.

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font
- Enable auto-kerning with `nil` or `[NSNull null]`
- Disable kerning – `@0.0`
- Any other value adjusts the kerning by a number of points

```
@{NSKernAttributeName: @2.0}
```

Big *red* dog.

Advanced Attributes

NSKernAttributeName

- Kerning will vary from font to font
- Enable auto-kerning with `nil` or `[NSNull null]`
- Disable kerning – `@0.0`
- Any other value adjusts the kerning by a number of points

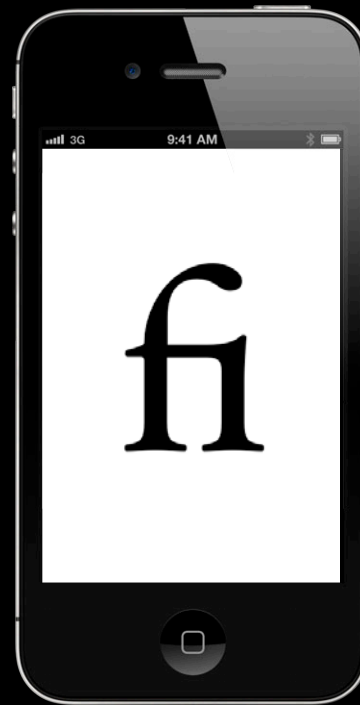
```
@{NSKernAttributeName: @-2.0}
```



Big *red* dog.

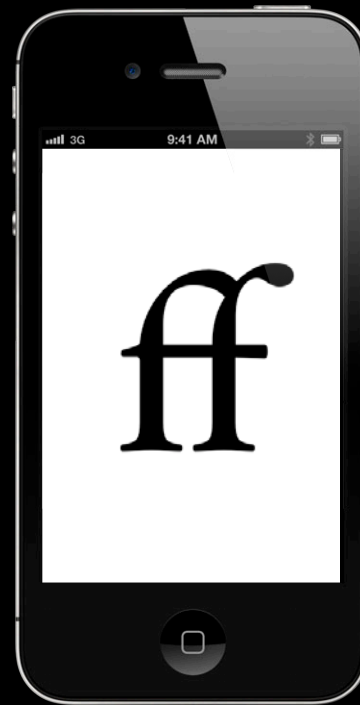
Advanced Attributes

NSLigatureAttributeName



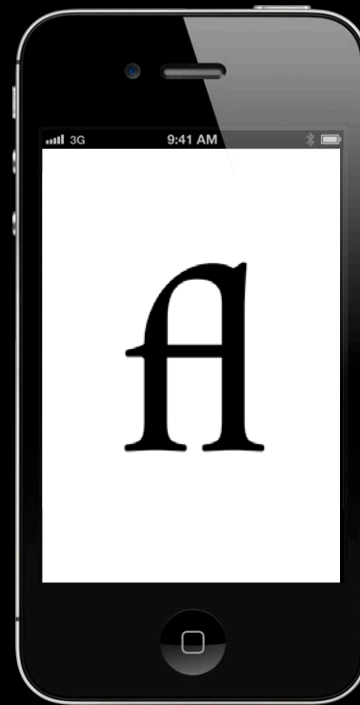
Advanced Attributes

NSLigatureAttributeName



Advanced Attributes

NSLigatureAttributeName



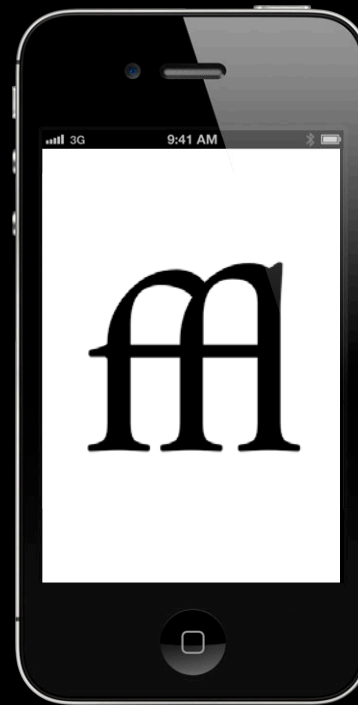
Advanced Attributes

NSLigatureAttributeName



Advanced Attributes

NSLigatureAttributeName



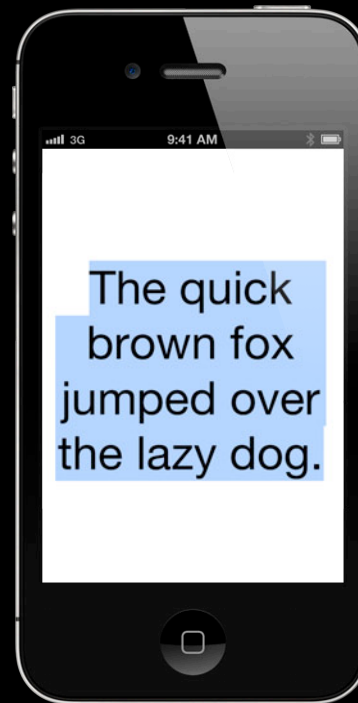
Advanced Attributes

`NSBackgroundColorAttributeName`



Advanced Attributes

`NSBackgroundColorAttributeName`



Advanced Attributes

`NSUnderlineStyleAttributeName`



Advanced Attributes

NSUnderlineStyleAttributeName



Advanced Attributes

`NSStrikethroughStyleAttributeName`



Advanced Attributes

NSStrikethroughStyleAttributeName



NSParagraphStyle

Line height

- Controls the line height

NSParagraphStyle

Line height

- Controls the line height
 - `minimumLineHeight` / `maximumLineHeight`

NSParagraphStyle

Line height

- Controls the line height
 - `minimumLineHeight` / `maximumLineHeight`
 - Usually set to same value

NSParagraphStyle

Line height

- Controls the line height
 - `minimumLineHeight` / `maximumLineHeight`
 - Usually set to same value
 - Moves baseline

NSParagraphStyle

Line height

- Controls the line height
 - `minimumLineHeight` / `maximumLineHeight`
 - Usually set to same value
 - Moves baseline
 - Used by Stocks



NSParagraphStyle

Spacing

- Controls the gap between lines and paragraphs
 - `lineSpacing`
 - a.k.a. “leading”
 - included in line fragment heights

NSParagraphStyle

Spacing

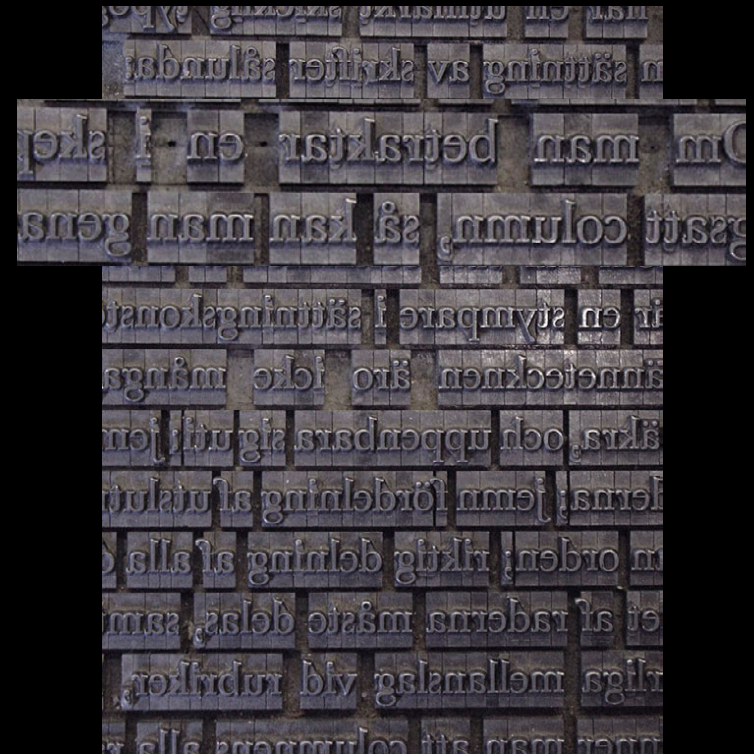
- Controls the gap between lines and paragraphs
 - lineSpacing
 - a.k.a. “leading”
 - included in line fragment heights



NSParagraphStyle

Spacing

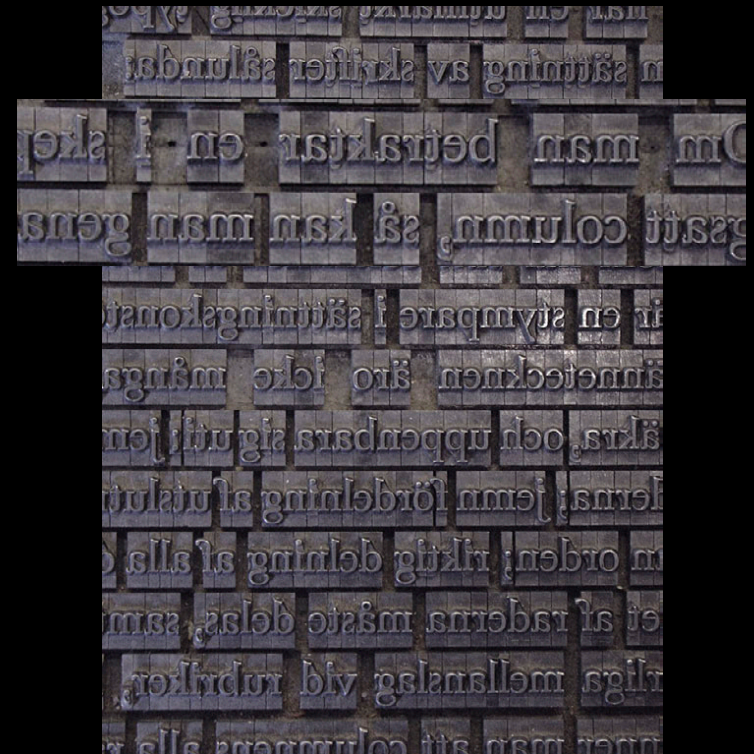
- Controls the gap between lines and paragraphs
 - `lineSpacing`



NSParagraphStyle

Spacing

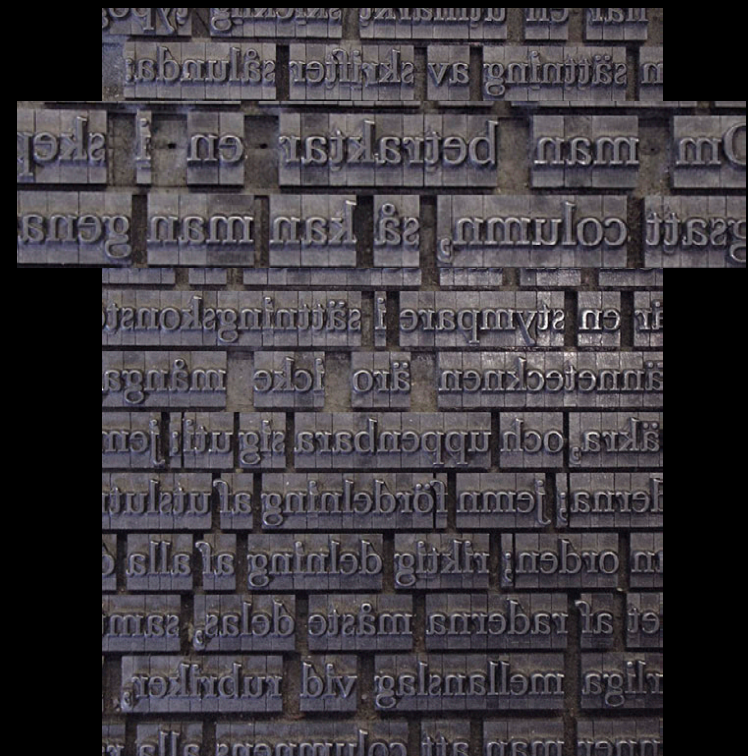
- Controls the gap between lines and paragraphs
 - lineSpacing
 - a.k.a. "leading"



NSParagraphStyle

Spacing

- Controls the gap between lines and paragraphs
 - `lineSpacing`
 - a.k.a. “leading”
 - included in line fragment heights



NSParagraphStyle

Spacing

- Controls the gap between lines in the paragraph
 - `lineSpacing`


This paragraph is using the normal line spacing value.

NSParagraphStyle

Spacing

- Controls the gap between lines in the paragraph
 - `lineSpacing`

The line spacing value for this
paragraph is 30.0.

A red vertical line with a horizontal bar at the top and bottom is positioned between the two lines of text. A red arrow points from the number '30.0' to the right side of this vertical line, indicating the distance between the two lines of text.

NSParagraphStyle

More spacing

- Controls the gap between lines and paragraphs
 - `paragraphSpacing`

This paragraph has 30.0 paragraph spacing.

← 30.0

This paragraph has 0 spacing values.

NSParagraphStyle

More spacing

- Controls the gap between lines and paragraphs
 - `paragraphSpacingBefore`

This paragraph has 0 spacing values.

← 30.0

The spacing value before this paragraph is 30.0.

NSParagraphStyle

Indentation

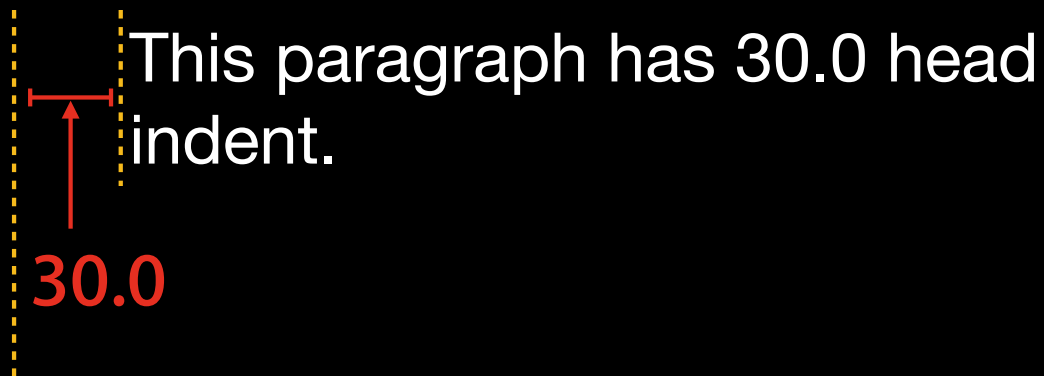
- Controls left and right margins
 - headIndent

This is a normal paragraph that has no indentation specified.

NSParagraphStyle

Indentation

- Controls left and right margins
 - headIndent

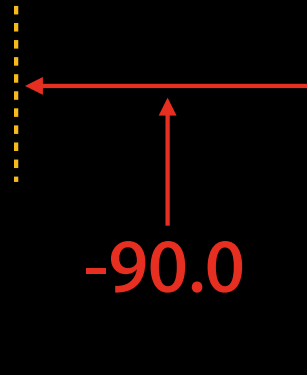


NSParagraphStyle

Indentation

- Controls left and right margins
 - tailIndent

This paragraph has 90.0 tail indent.

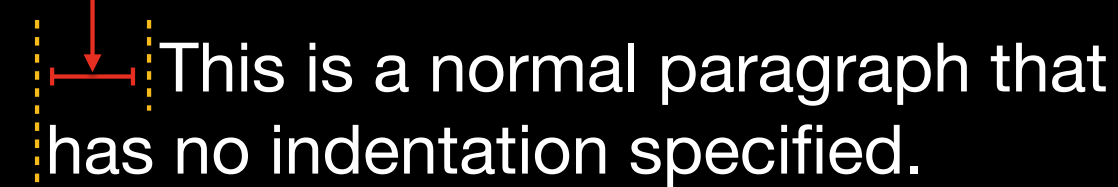


NSParagraphStyle

Indentation

- Controls left and right margins
 - `firstLineHeadIndent`

30.0



This is a normal paragraph that
has no indentation specified.

NSParagraphStyle

Hyphenation

NSParagraphStyle

Hyphenation

- Control hyphenation

NSParagraphStyle

Hyphenation

- Control hyphenation
 - `hyphenationFactor`

NSParagraphStyle

Hyphenation

- Control hyphenation
 - hyphenationFactor
 - Disabled—`0.0f`

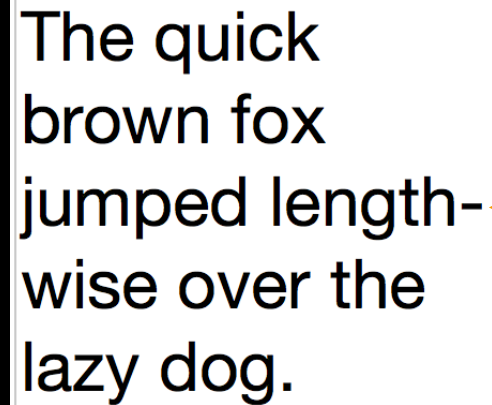
The quick
brown fox
jumped
lengthwise over
the lazy dog.

NSParagraphStyle

Hyphenation

- Control hyphenation
 - hyphenationFactor
 - Disabled—`0.0f`
 - Enabled—`1.0f`

The quick
brown fox
jumped length-
wise over the
lazy dog.



Text Effects in Your App

Text Effects in Your App

Using advanced text effects to beautify and enhance

John **Appleseed**

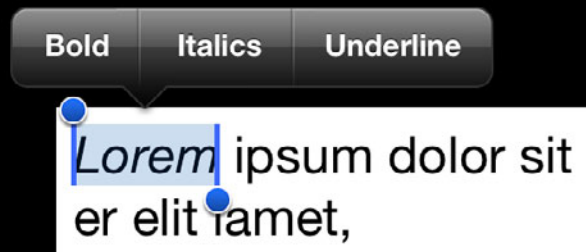
Text Effects in Your App

Using advanced text effects to beautify and enhance

UITextField!

Text Effects in Your App

Using advanced text effects to beautify and enhance



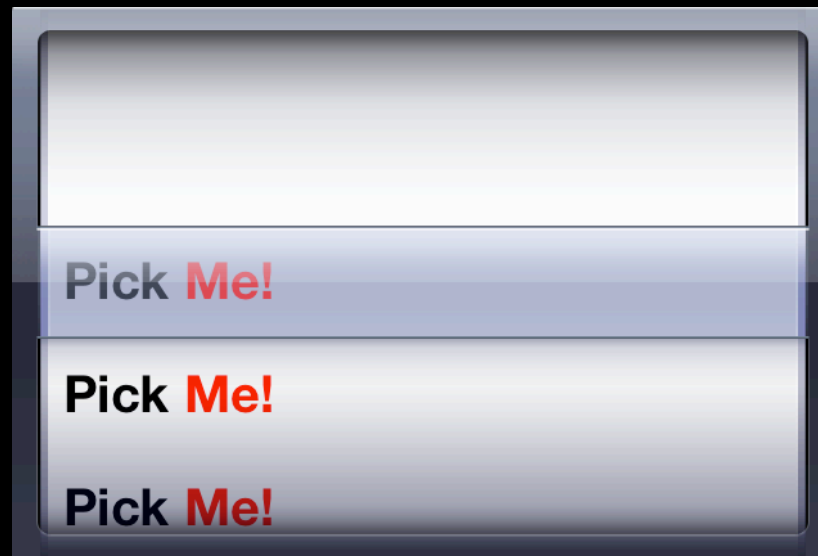
Text Effects in Your App

Using advanced text effects to beautify and enhance

Touch **Me**

Text Effects in Your App

Using advanced text effects to beautify and enhance



Text Effects in Your App

UIKit caveats

Text Effects in Your App

UIKit caveats

- UIKit supplies default values for

Text Effects in Your App

UIKit caveats

- UIKit supplies default values for

NSFontAttributeName — [UIFont systemFontOfSize:DEFAULT_FONT_SIZE]

Text Effects in Your App

UIKit caveats

- UIKit supplies default values for

NSFontAttributeName — [UIFont systemFontOfSize:DEFAULT_FONT_SIZE]

NSForegroundColorAttributeName — [UIColor blackColor]

Text Effects in Your App

UIKit caveats

- UIKit supplies default values for

NSFontAttributeName — [UIFont systemFontOfSize:DEFAULT_FONT_SIZE]

NSForegroundColorAttributeName — [UIColor blackColor]

NSKernAttributeName — @0.0

Text Effects in Your App

UIKit caveats

- UIKit supplies default values for

NSFontAttributeName — [UIFont systemFontOfSize:DEFAULT_FONT_SIZE]

NSForegroundColorAttributeName — [UIColor blackColor]

NSKernAttributeName — @0.0

NSLigatureAttributeName — @0

Text Effects in Your App

Table Views



Text Effects in Your App

Table Views

UITableViewHeaderFooterView
.textLabel
.attributedString



Text Effects in Your App

Table Views

UITableViewHeaderFooterView
.textLabel
.attributedString

NSShadowAttributeName



Text Effects in Your App

Table Views



`UITableViewCell`
`.textLabel`
`.attributedText`
`NSKernAttributeName`

Text Effects in Your App

Table Views



`UITableViewCell`
`.textLabel`
`.attributedString`
`NSKernAttributeName`
`NSLigatureAttributeName`

Contexts and Options

NSStringDrawing API



- Simple drawing/sizing methods
 - `(void)drawInRect:(CGRect)rect;`
 - `(void)drawAtPoint:(CGPoint)point;`
 - `(CGSize)size;`
- Extended drawing/sizing methods
 - `(void)drawWithRect:(CGRect)rect`
`options:(NSStringDrawingOptions)options`
`context:(NSStringDrawingContext *)context;`
 - `(CGRect)boundingRectWithSize:(CGSize)size`
`options:(NSStringDrawingOptions)options`
`context:(NSStringDrawingContext *)context;`

NSStringDrawing API



- Simple drawing/sizing methods
 - `(void)drawInRect:(CGRect)rect;`
 - `(void)drawAtPoint:(CGPoint)point;`
 - `(CGSize)size;`
- Extended drawing/sizing methods
 - `(void)drawWithRect:(CGRect)rect`
`options:(NSStringDrawingOptions)options`
`context:(NSStringDrawingContext *)context;`
 - `(CGRect)boundingRectWithSize:(CGSize)size`
`options:(NSStringDrawingOptions)options`
`context:(NSStringDrawingContext *)context;`

Extended NSStringDrawing

Extended NSStringDrawing

- Additional drawing/sizing capabilities

Extended NSStringDrawing

- Additional drawing/sizing capabilities
- Baseline vs. line fragment origin layout

Extended NSStringDrawing

- Additional drawing/sizing capabilities
- Baseline vs. line fragment origin layout
- Specify the constraining size

NSStringDrawingOptions

Values

- Supported values

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

`NSStringDrawingTruncatesLastVisibleLine`

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

`NSStringDrawingTruncatesLastVisibleLine`

`NSStringDrawingUsesFontLeading`

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

`NSStringDrawingTruncatesLastVisibleLine`

`NSStringDrawingUsesFontLeading`

- Specifying multiple options

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

`NSStringDrawingTruncatesLastVisibleLine`

`NSStringDrawingUsesFontLeading`

- Specifying multiple options

- Truncate last visible line and use line fragment origin

NSStringDrawingOptions

Values

- Supported values

`NSStringDrawingUsesLineFragmentOrigin`

`NSStringDrawingTruncatesLastVisibleLine`

`NSStringDrawingUsesFontLeading`

- Specifying multiple options

- Truncate last visible line and use line fragment origin

- `NSStringDrawingTruncatesLastVisibleLine | NSStringDrawingUsesLineFragmentOrigin`

Origin

NSStringDrawingUsesLineFragmentOrigin

Origin

NSStringDrawingUsesLineFragmentOrigin

- The origin of the rect determines where the rendering begins

Origin

NSStringDrawingUsesLineFragmentOrigin

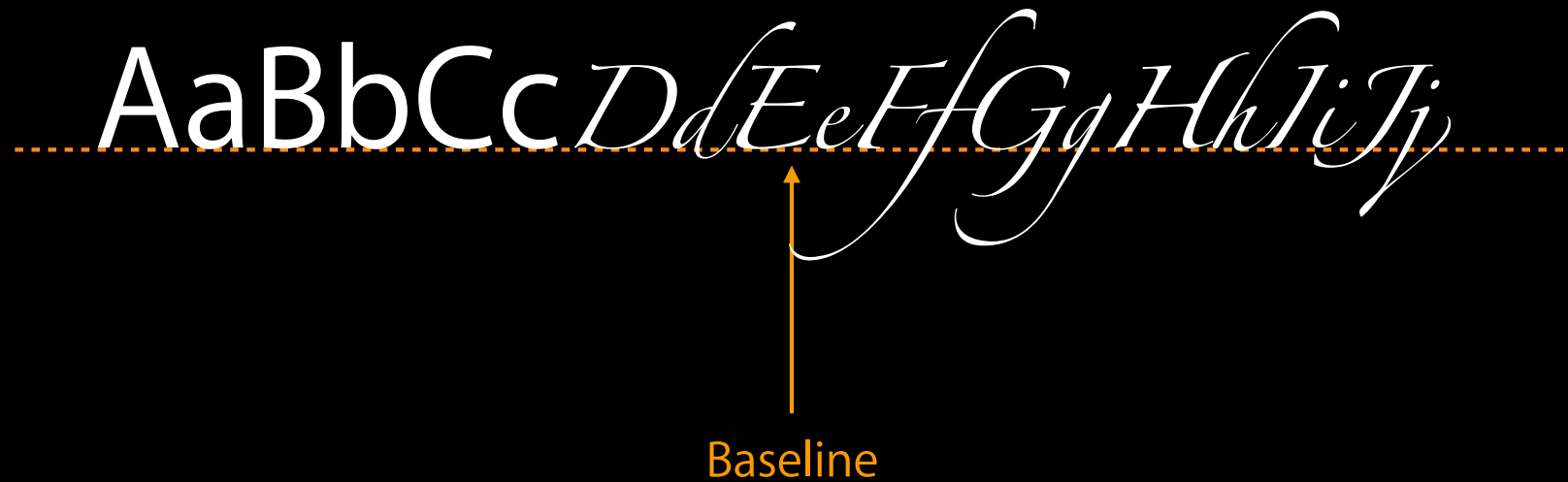
- The origin of the rect determines where the rendering begins
 - (void)drawWithRect:(CGRect)rect options:context:

Baseline Origin

NSStringDrawingUsesLineFragmentOrigin

Baseline Origin

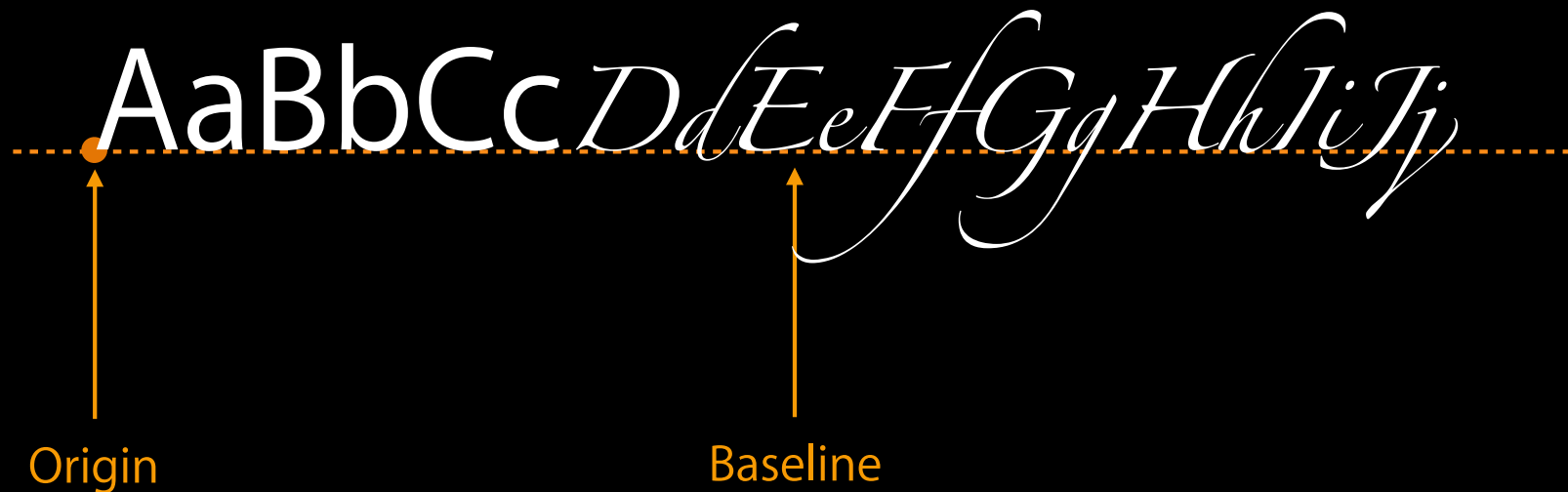
NSStringDrawingUsesLineFragmentOrigin



Baseline Origin

NSStringDrawingUsesLineFragmentOrigin

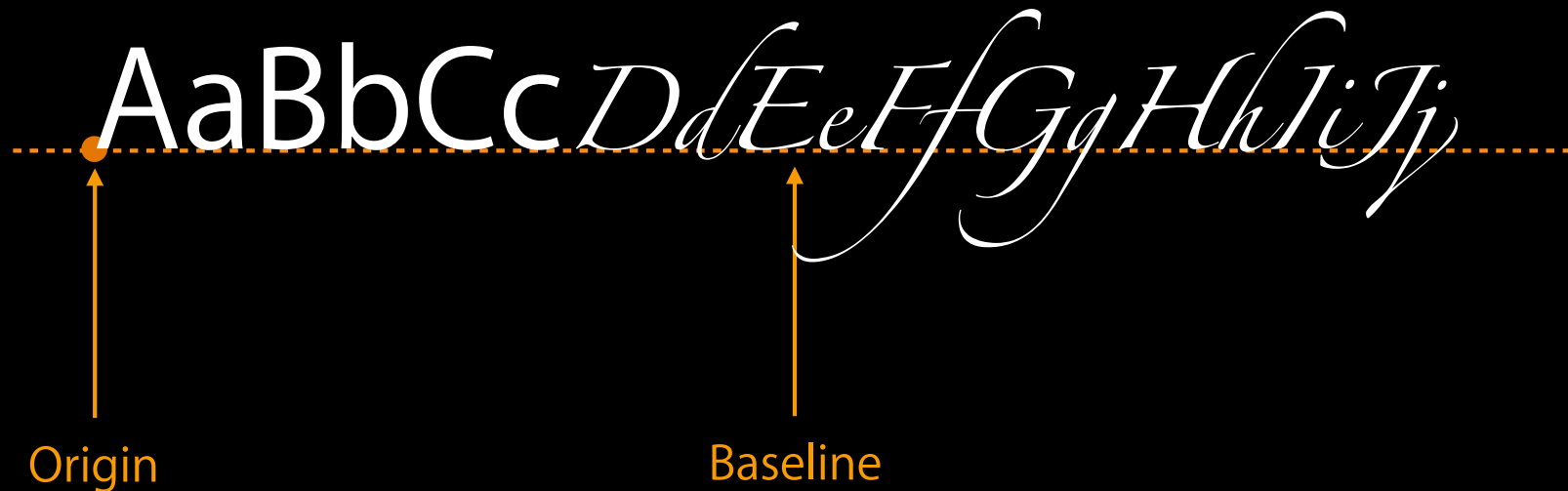
- Origin at baseline



Baseline Origin

NSStringDrawingUsesLineFragmentOrigin

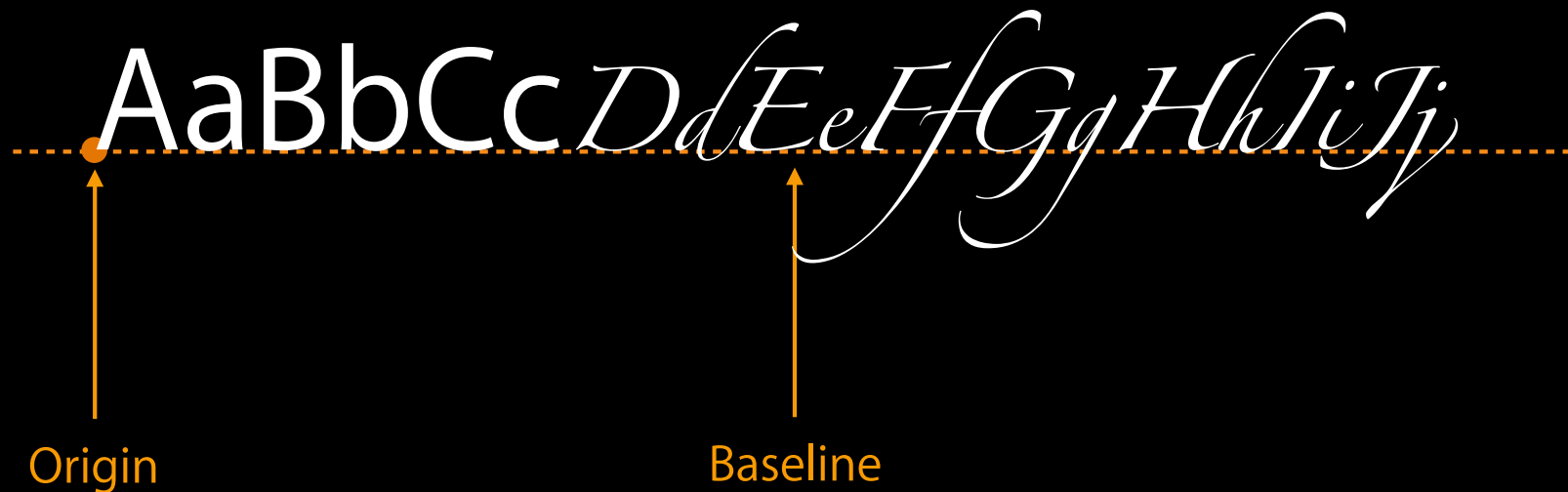
- Origin at baseline
- Default mode for extended API



Baseline Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at baseline
- Default mode for extended API
- Usually used with a single line of text

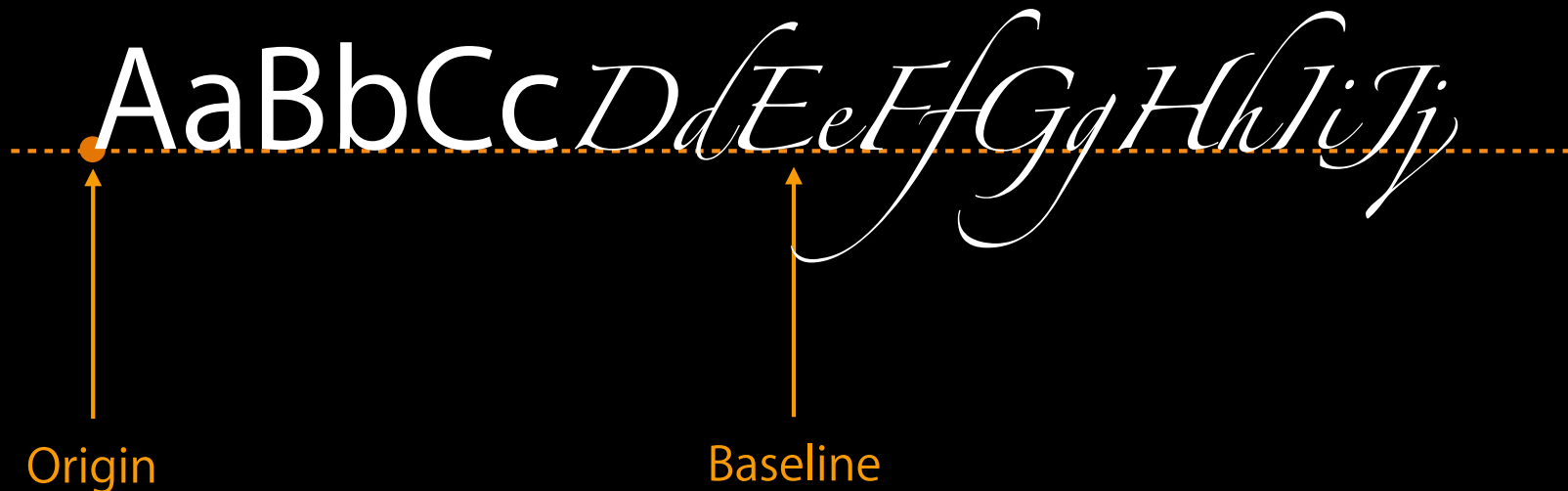


Baseline Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at baseline
- Default mode for extended API
- Usually used with a single line of text

NSStringDrawingUsesLineFragmentOrigin **not** set



Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin

AaBbCcDdEe
FfGgHhIiJj

Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin



Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at upper-left of line fragment



Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at upper-left of line fragment
- Default mode for simple API



Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at upper-left of line fragment
- Default mode for simple API
- Usually used with multiple lines of text



Line Fragment Origin

NSStringDrawingUsesLineFragmentOrigin

- Origin at upper-left of line fragment
- Default mode for simple API
- Usually used with multiple lines of text



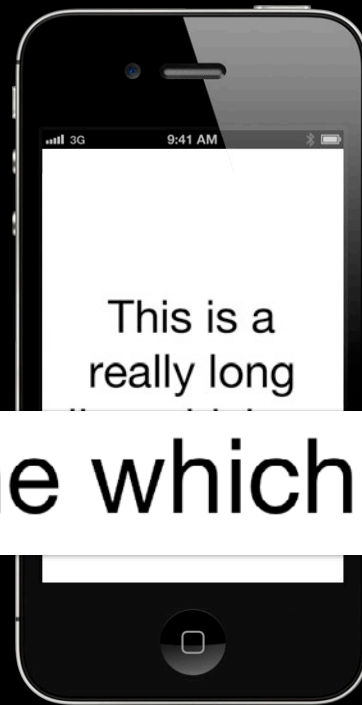
Controlling Truncation

`NSStringDrawingTruncatesLastVisibleLine`



Controlling Truncation

`NSStringDrawingTruncatesLastVisibleLine`



line which...

NSStringDrawingContext

Setup



NSStringDrawingContext

Setup

- Read-write properties (for you to setup):



Big red dog.

NSStringDrawingContext

Setup

- Read-write properties (for you to setup):

`minimumScaleFactor`



Big red ...

NSStringDrawingContext

Setup

- Read-write properties (for you to setup):

`minimumScaleFactor`

`minimumTrackingAdjustment`



Big red dog.

NSStringDrawingContext

Data

- Read-only properties

NSStringDrawingContext

Data

- Read-only properties

`actualScaleFactor`

Big red ... ← 50%

NSStringDrawingContext

Data

- Read-only properties

`actualScaleFactor`

`actualTrackingAdjustment`

Big red dog. ← 0.15

NSStringDrawingContext

Data

- Read-only properties

`actualScaleFactor`

`actualTrackingAdjustment`

`totalBounds`



NSStringDrawingContext

Consistent font label sizes and tracking

NSStringDrawingContext

Consistent font label sizes and tracking

scaleFactor = 0.8

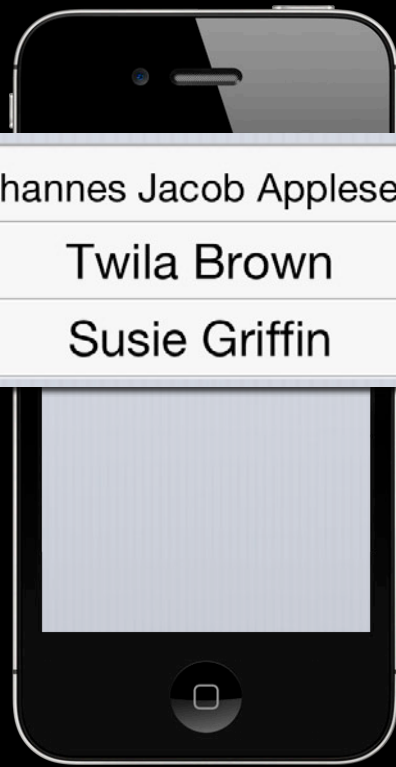
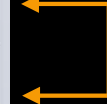


Johannes Jacob Appleseed

Twila Brown

Susie Griffin

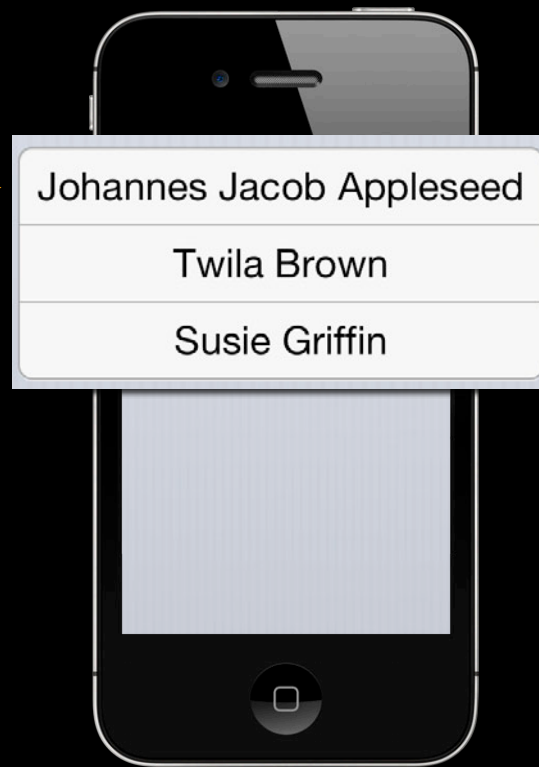
scaleFactor = 1.0



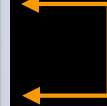
NSStringDrawingContext

Consistent font label sizes and tracking

scaleFactor = 0.8



scaleFactor = 0.8



NSStringDrawingContext

One-pass layout

NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31
Road Trip
February 9, 2012
1,664,219 views  86%



Apple - iPhone 4S - TV Ad - 00:31
Santa
December 18, 2011
3,197,926 views  95%



Apple - iPhone 4S - TV Ad - 00:31
iCloud
October 30, 2011
931,458 views  90%



Apple - iPhone 4S - TV Ad - 00:30
Camera
October 30, 2011
1,470,900 views  90%

NSStringDrawingContext

One-pass layout



NSStringDrawingContext

One-pass layout



00:31

NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad -
Road Trip

00:31

NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31
Road Trip

February 9, 2012

NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31

Road Trip

February 9, 2012

1,664,219 views

NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31
Road Trip

February 9, 2012
1,664,219 views



NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31

Road Trip

February 9, 2012

1,664,219 views



NSStringDrawingContext

One-pass layout



Apple - iPhone 4S - TV Ad - 00:31
Road Trip
February 9, 2012
1,664,219 views  86%



Apple - iPhone 4S - TV Ad - 00:31
Santa
December 18, 2011
3,197,926 views  95%



Apple - iPhone 4S - TV Ad - 00:31
iCloud
October 30, 2011
931,458 views  90%



Apple - iPhone 4S - TV Ad - 00:30
Camera
October 30, 2011
1,470,900 views  90%

NSStringDrawing

Concurrency

NSStringDrawing

Concurrency

- NSStringDrawing calls are thread safe



NSStringDrawing

Concurrency

- NSStringDrawing calls are thread safe
- NSStringDrawingContext instances are not
 - No sharing instances between queues



Demo

Johannes Fortmann
UIKit Engineer

Demo Recap

Demo Recap

- Use advanced text effects to communicate emphasis, selection

Demo Recap

- Use advanced text effects to communicate emphasis, selection
- Make your text more legible

Demo Recap

- Use advanced text effects to communicate emphasis, selection
- Make your text more legible
- Drawing on the background works

Demo Recap

- Use advanced text effects to communicate emphasis, selection
- Make your text more legible
- Drawing on the background works
 - Use `UIImage` and `drawInRect:`

Demo Recap

- Use advanced text effects to communicate emphasis, selection
- Make your text more legible
- Drawing on the background works
 - Use `UIImage` and `drawInRect:`
 - Do not manipulate UI elements from the background queue

Summary

Summary

- Advanced attributes

Summary

- Advanced attributes
- Text effects in your App

Summary

- Advanced attributes
- Text effects in your App
- Contexts and options

More Information

Jake Behrens

UI Frameworks Evangelist
behrens@apple.com

Documentation

UIKit Framework Reference
<http://developer.apple.com/library/ios/>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Introduction to Attributed Strings for iOS

Mission
Wednesday 3:15PM

Core Text and Fonts

Russian Hill
Wednesday 4:30PM

Keyboard Input in iOS

Russian Hill
Wednesday 2:00PM

Labs

Attributed Strings & Text Lab

Essentials Lab A
Thursday 11:30AM

 **WWDC2012**

