

---

# NSString Application Kit Additions Reference

[Cocoa > Data Management](#)



2009-02-04



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **NSString Application Kit Additions Reference 5**

---

Overview 5

Tasks 5

    Drawing String Objects 5

    Getting the Bounding Rect of Rendered Strings 6

Instance Methods 6

    boundingRectWithSize:options:attributes: 6

    drawAtPoint:withAttributes: 6

    drawInRect:withAttributes: 7

    drawWithRect:options:attributes: 8

    sizeWithAttributes: 9

Constants 9

    String Drawing Options 9

---

## **Document Revision History 11**

---

## **Index 13**

---



# NSString Application Kit Additions Reference

---

<b>Inherits from</b>	NSObject
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Declared in</b>	NSStringDrawing.h
<b>Companion guides</b>	String Programming Guide for Cocoa Attributed Strings Programming Guide

## Overview

The Application Kit adds three methods to the `NSString` class to support drawing string objects directly in an `NSView` instance: `drawAtPoint:withAttributes:` (page 6), `drawInRect:withAttributes:` (page 7), and `sizeWithAttributes:` (page 9).

The Application Kit adds similar methods to the `NSAttributedString` class, described in *NSAttributedString Application Kit Additions Reference*. The two drawing methods draw a string object with a single set of attributes that apply to the entire string. To draw a string with multiple attributes, such as multiple text fonts, you must use an attributed string.

## Tasks

### Drawing String Objects

- `drawAtPoint:withAttributes:` (page 6)  
Draws the receiver with the font and other display characteristics of the given attributes, at the specified point in the currently focused view.
- `drawInRect:withAttributes:` (page 7)  
Draws the receiver with the font and other display characteristics of the given attributes, within the specified rectangle in the currently focused `NSView`.
- `drawWithRect:options:attributes:` (page 8)  
Draws the receiver with the specified options and other display characteristics of the given attributes, within the specified rectangle in the current graphics context.
- `sizeWithAttributes:` (page 9)  
Returns the bounding box size the receiver occupies when drawn with the given attributes.

## Getting the Bounding Rect of Rendered Strings

- [boundingRectWithSize:options:attributes:](#) (page 6)

Calculates and returns the bounding rect for the receiver drawn using the given options and display characteristics, within the specified rectangle in the current graphics context.

## Instance Methods

### **boundingRectWithSize:options:attributes:**

Calculates and returns the bounding rect for the receiver drawn using the given options and display characteristics, within the specified rectangle in the current graphics context.

```
- (NSRect)boundingRectWithSize:(NSSize)size options:(NSStringDrawingOptions)options
  attributes:(NSDictionary *)attributes
```

#### **Parameters**

*size*

The size of the rectangle to draw in.

*options*

String drawing options.

*attributes*

A dictionary of text attributes to be applied to the string. These are the same attributes that can be applied to an `NSAttributedString` object, but in the case of `NSString` objects, the attributes apply to the entire string, rather than ranges within the string.

#### **Return Value**

The bounding rect for the receiver drawn using the given options and display characteristics. The rect origin returned from this method is the first glyph origin.

#### **Availability**

Available in Mac OS X v10.4 and later.

#### **See Also**

[drawInRect:withAttributes:](#) (page 7)

#### **Declared In**

NSStringDrawing.h

### **drawAtPoint:withAttributes:**

Draws the receiver with the font and other display characteristics of the given attributes, at the specified point in the currently focused view.

```
- (void)drawAtPoint:(NSPoint)aPoint withAttributes:(NSDictionary *)attributes
```

**Parameters***aPoint*

The origin for the bounding box for drawing the string. If the focused view is flipped, the origin is the upper-left corner of the drawing bounding box; otherwise, the origin is the lower-left corner.

*attributes*

A dictionary of text attributes to be applied to the string. These are the same attributes that can be applied to an `NSAttributedString` object, but in the case of `NSString` objects, the attributes apply to the entire string, rather than ranges within the string.

**Discussion**

The width (height for vertical layout) of the rendering area is unlimited, unlike [drawInRect:withAttributes:](#) (page 7), which uses a bounding rectangle. As a result, this method renders the text in a single line.

You should only invoke this method when an `NSView` object has focus.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

[drawInRect:withAttributes:](#) (page 7) (`NSView`)

[drawWithRect:options:attributes:](#) (page 8)

- `lockFocus`

**Related Sample Code**

`CIVideoDemoGL`

`Clock Control`

`CocoaVideoFrameToGWorld`

`SpecialPictureProtocol`

**Declared In**

`NSStringDrawing.h`

**drawInRect:withAttributes:**

Draws the receiver with the font and other display characteristics of the given attributes, within the specified rectangle in the currently focused `NSView`.

```
- (void)drawInRect:(NSRect)aRect withAttributes:(NSDictionary *)attributes
```

**Parameters***aRect*

The rectangle in which to draw the string.

*attributes*

A dictionary of text attributes to be applied to the string. These are the same attributes that can be applied to an `NSAttributedString` object, but in the case of `NSString` objects, the attributes apply to the entire string, rather than ranges within the string.

**Discussion**

The rendering area is bounded by *aRect*, unlike [drawAtPoint:withAttributes:](#) (page 6), which has an unlimited width. As a result, this method renders the text in multiple lines.

You should only invoke this method when an `NSView` has focus.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

[drawAtPoint:withAttributes:](#) (page 6) (NSView)

[drawWithRect:options:attributes:](#) (page 8)

- lockFocus

**Related Sample Code**

FilterDemo

NURBSSurfaceVertexProg

VertexPerformanceDemo

VertexPerformanceTest

Worm

**Declared In**

NSStringDrawing.h

**drawWithRect:options:attributes:**

Draws the receiver with the specified options and other display characteristics of the given attributes, within the specified rectangle in the current graphics context.

```
- (void)drawWithRect:(NSRect)rect options:(NSStringDrawingOptions)options
    attributes:(NSDictionary *)attributes
```

**Parameters**

*rect*

The rectangle in which to draw the string.

*options*

String drawing options.

*attributes*

A dictionary of text attributes to be applied to the string. These are the same attributes that can be applied to an NSAttributedString object, but in the case of NSString objects, the attributes apply to the entire string, rather than ranges within the string.

**Discussion**

The *rect* argument's *origin* field specifies the rendering origin. The point is interpreted as the baseline origin by default. With `NSStringDrawingUsesLineFragmentOrigin`, it is interpreted as the upper left corner of the line fragment rect. The *size* field specifies the text container size. The *width* part of the *size* field specifies the maximum line fragment width if larger than 0.0. The *height* defines the maximum size that can be occupied with text if larger than 0.0 and `NSStringDrawingUsesLineFragmentOrigin` is specified. If `NSStringDrawingUsesLineFragmentOrigin` is not specified, *height* is ignored and considered to be single-line rendering (`NSLineBreakByWordWrapping` and `NSLineBreakByCharWrapping` are treated as `NSLineBreakByClipping`).

You should only invoke this method when there is a current graphics context.

**Availability**

Available in Mac OS X v10.4 and later.



**See Also**[drawAtPoint:withAttributes:](#) (page 6) (NSView)[drawInRect:withAttributes:](#) (page 7)

- lockFocus

**Declared In**

NSStringDrawing.h

**sizeWithAttributes:**

Returns the bounding box size the receiver occupies when drawn with the given attributes.

- (NSSize)sizeWithAttributes:(NSDictionary \*)*attributes***Parameters***attributes*

A dictionary of text attributes to be applied to the string. These are the same attributes that can be applied to an NSAttributedString object, but in the case of NSString objects, the attributes apply to the entire string, rather than ranges within the string.

**Return Value**The bounding box size the receiver occupies when drawn with *attributes*.**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

CIVideoDemoGL

CocoaVideoFrameToGWorld

SpecialPictureProtocol

**Declared In**

NSStringDrawing.h

## Constants

### String Drawing Options

The following constants are provided as rendering options for a string when it is drawn.

```
enum {
    NSStringDrawingUsesLineFragmentOrigin = (1 << 0),
    NSStringDrawingUsesFontLeading = (1 << 1),
    NSStringDrawingDisableScreenFontSubstitution = (1 << 2),
    NSStringDrawingUsesDeviceMetrics = (1 << 3),
    NSStringDrawingOneShot = (1 << 4),
    NSStringDrawingTruncatesLastVisibleLine = (1 << 5)
};
typedef NSInteger NSStringDrawingOptions;
```

### Constants

`NSStringDrawingUsesLineFragmentOrigin`

The specified origin is the line fragment origin, not the baseline origin.

Available in Mac OS X v10.4 and later.

Declared in `NSStringDrawing.h`.

`NSStringDrawingUsesFontLeading`

Uses the font leading for calculating line heights.

Available in Mac OS X v10.4 and later.

Declared in `NSStringDrawing.h`.

`NSStringDrawingDisableScreenFontSubstitution`

Disable screen font substitution (equivalent to `[NSLayoutManager setUsesScreenFonts:NO]`).

Available in Mac OS X v10.4 and later.

Declared in `NSStringDrawing.h`.

`NSStringDrawingUsesDeviceMetrics`

Uses image glyph bounds instead of typographic bounds.

Available in Mac OS X v10.4 and later.

Declared in `NSStringDrawing.h`.

`NSStringDrawingOneShot`

Suppresses caching layout information.

Available in Mac OS X v10.4 and later.

Declared in `NSStringDrawing.h`.

`NSStringDrawingTruncatesLastVisibleLine`

Truncates and adds the ellipsis character to the last visible line if the text doesn't fit into the bounds specified.

This option is ignored if `NSStringDrawingUsesLineFragmentOrigin` is not also set. In addition, the line break mode must be either `NSLineBreakByWordWrapping` or `NSLineBreakByCharWrapping` for this option to take effect. The line break mode can be specified in a paragraph style passed in the attributes dictionary argument of the drawing methods.

Available in Mac OS X v10.5 and later.

Declared in `NSStringDrawing.h`.

# Document Revision History

---

This table describes the changes to *NSString Application Kit Additions Reference*.

Date	Notes
2009-02-04	Documented NSStringDrawingTruncatesLastVisibleLine constant.
2006-06-28	Added parameter descriptions.
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History

# Index

---

## B

---

`boundingRectWithSize:options:attributes:`  
instance method [6](#)

## D

---

`drawAtPoint:withAttributes:` instance method [6](#)  
`drawInRect:withAttributes:` instance method [7](#)  
`drawWithRect:options:attributes:` instance  
method [8](#)

## N

---

`NSStringDrawingDisableScreenFontSubstitution`  
constant [10](#)  
`NSStringDrawingOneShot` constant [10](#)  
`NSStringDrawingTruncatesLastVisibleLine`  
constant [10](#)  
`NSStringDrawingUsesDeviceMetrics` constant [10](#)  
`NSStringDrawingUsesFontLeading` constant [10](#)  
`NSStringDrawingUsesLineFragmentOrigin` constant  
[10](#)

## S

---

`sizeWithAttributes:` instance method [9](#)  
String Drawing Options [9](#)