

---

# NSAttributedString Class Reference

[Cocoa > Data Management](#)



2008-10-15



Apple Inc.  
© 2008 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.

Aperture and Spotlight are trademarks of Apple Inc.

Helvetica is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

---

## **NSAttributedString Class Reference 5**

---

Overview	5
Adopted Protocols	6
Tasks	6
Creating an NSAttributedString Object	6
Retrieving Character Information	6
Retrieving Attribute Information	7
Comparing Attributed Strings	7
Extracting a Substring	7
Instance Methods	7
attributeAtIndex:effectiveRange:	7
attributeAtIndex:longestEffectiveRange:inRange:	8
attributedStringFromRange:	9
attributesAtIndex:effectiveRange:	10
attributesAtIndex:longestEffectiveRange:inRange:	10
initWithAttributedString:	11
initWithString:	11
initWithString:attributes:	12
isEqualToAttributedString:	12
length	13
string	13
Constants	14

---

## **Document Revision History 15**

---

## **Index 17**

---



# NSAttributedString Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying NSMutableCopying NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Attributed Strings Programming Guide
<b>Declared in</b>	NSAttributedString.h
<b>Related sample code</b>	CIAnnotation CoreRecipes iSpend OpenGL Screensaver StickiesExample

## Overview

`NSAttributedString` objects manage character strings and associated sets of attributes (for example, font and kerning) that apply to individual characters or ranges of characters in the string. An association of characters and their attributes is called an attributed string. The cluster's two public classes, `NSAttributedString` and `NSMutableAttributedString`, declare the programmatic interface for read-only attributed strings and modifiable attributed strings, respectively. The Foundation framework defines only the basic functionality for attributed strings; additional methods supporting RTF, graphics attributes, and drawing attributed strings are described in `NSAttributedString Additions`, found in the Application Kit. The Application Kit also uses a subclass of `NSMutableAttributedString`, called `NSTextStorage`, to provide the storage for the Application Kit's extended text-handling system.

The Application Kit also uses `NSParagraphStyle` and its subclass `NSMutableParagraphStyle` to encapsulate the paragraph or ruler attributes used by the `NSAttributedString` classes.

An attributed string identifies attributes by name, storing a value under the name in an `NSDictionary` object. Standard attribute keys are described in the "Constants" section of *NSAttributedString Application Kit Additions Reference*. You can also assign any attribute name/value pair you wish to a range of characters—it is up to your application to interpret custom attributes (see *Attributed Strings Programming Guide*).

Note that the default font for `NSAttributedString` objects is Helvetica 12-point, which differs from the Mac OS X system font Lucida Grande, so you may wish to create the string with non-default attributes suitable for your application using, for example, `initWithString:attributes:` (page 12).

Be aware that `isEqual:` comparison among `NSAttributedString` objects compares for exact equality, including not only literal character-by-character string equality but also equality of all attributes, which is not likely to be achieved in the case of many attributes such as attachments, lists, and tables, for example.

## Adopted Protocols

### NSCoding

`encodeWithCoder:`

`initWithCoder:`

### NSCopying

`copyWithZone:`

### NSMutableCopying

`mutableCopyWithZone:`

## Tasks

### Creating an NSAttributedString Object

- `initWithString:` (page 11)  
Returns an `NSAttributedString` object initialized with the characters of a given string and no attribute information.
- `initWithAttributedString:` (page 11)  
Returns an `NSAttributedString` object initialized with the characters and attributes of another given attributed string.
- `initWithString:attributes:` (page 12)  
Returns an `NSAttributedString` object initialized with a given string and attributes.

### Retrieving Character Information

- `string` (page 13)  
Returns the character contents of the receiver as an `NSString` object.
- `length` (page 13)  
Returns the length of the receiver's string object.

## Retrieving Attribute Information

- [attributesAtIndex:effectiveRange:](#) (page 10)  
Returns the attributes for the character at a given index.
- [attributesAtIndex:longestEffectiveRange:inRange:](#) (page 10)  
Returns the attributes for the character at a given index, and by reference the range over which the attributes apply.
- [attribute:atIndex:effectiveRange:](#) (page 7)  
Returns the value for an attribute with a given name of the character at a given index, and by reference the range over which the attribute applies.
- [attribute:atIndex:longestEffectiveRange:inRange:](#) (page 8)  
Returns the value for the attribute with a given name of the character at a given index, and by reference the range over which the attribute applies.

## Comparing Attributed Strings

- [isEqualToAttributedString:](#) (page 12)  
Returns a Boolean value that indicates whether the receiver is equal to another given attributed string.

## Extracting a Substring

- [attributedStringFromRange:](#) (page 9)  
Returns an NSAttributedString object consisting of the characters and attributes within a given range in the receiver.

## Instance Methods

### **attribute:atIndex:effectiveRange:**

Returns the value for an attribute with a given name of the character at a given index, and by reference the range over which the attribute applies.

```
(id)attribute:(NSString *)attributeName atIndex:(NSUInteger)index
effectiveRange:(NSRangePointer)aRange
```

#### Parameters

*attributeName*

The name of an attribute.

*index*

The index for which to return attributes. This value must not exceed the bounds of the receiver.

*aRange*

If non-NULL:

- If the named attribute exists at *index*, upon return *aRange* contains a range over which the named attribute's value applies.
- If the named attribute does not exist at *index*, upon return *aRange* contains the range over which the attribute does not exist.

The range isn't necessarily the maximum range covered by *attributeName*, and its extent is implementation-dependent. If you need the maximum range, use [attribute:atIndex:longestEffectiveRange:inRange:](#) (page 8). If you don't need this value, pass NULL.

#### Return Value

The value for the attribute named *attributeName* of the character at index *index*, or *nil* if there is no such attribute.

#### Discussion

Raises an `NSRangeException` if *index* lies beyond the end of the receiver's characters.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [attributesAtIndex:effectiveRange:](#) (page 10)

#### Related Sample Code

`iSpend`

`TextLinks`

#### Declared In

`NSAttributedString.h`

## **attribute:atIndex:longestEffectiveRange:inRange:**

Returns the value for the attribute with a given name of the character at a given index, and by reference the range over which the attribute applies.

```
- (id)attribute:(NSString *)attributeName atIndex:(NSUInteger)index
    longestEffectiveRange:(NSRangePointer)aRange inRange:(NSRange)rangeLimit
```

#### Parameters

*attributeName*

The name of an attribute.

*index*

The index at which to test for *attributeName*.



*aRange*

If non-NULL:

- If the named attribute exists at *index*, upon return *aRange* contains the full range over which the value of the named attribute is the same as that at *index*, clipped to *rangeLimit*.
- If the named attribute does not exist at *index*, upon return *aRange* contains the full range over which the attribute does not exist, clipped to *rangeLimit*.

If you don't need this value, pass NULL.

*rangeLimit*

The range over which to search for continuous presence of *attributeName*. This value must not exceed the bounds of the receiver.

### Return Value

The value for the attribute named *attributeName* of the character at *index*, or `nil` if there is no such attribute.

### Discussion

Raises an `NSRangeException` if *index* or any part of *rangeLimit* lies beyond the end of the receiver's characters.

If you don't need the longest effective range, it's far more efficient to use the [attribute:atIndex:effectiveRange:](#) (page 7) method to retrieve the attribute value.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [attributesAtIndex:longestEffectiveRange:inRange:](#) (page 10)

### Declared In

NSAttributedString.h

## attributedStringFromRange:

Returns an `NSAttributedString` object consisting of the characters and attributes within a given range in the receiver.

```
- (NSAttributedString *)attributedStringFromRange:(NSRange)aRange
```

### Parameters

*aRange*

The range from which to create a new attributed string. *aRange* must lie within the bounds of the receiver.

### Return Value

An `NSAttributedString` object consisting of the characters and attributes within *aRange* in the receiver.

### Discussion

Raises an `NSRangeException` if any part of *aRange* lies beyond the end of the receiver's characters. This method treats the length of the string as a valid range value that returns an empty string.

### Availability

Available in Mac OS X v10.0 and later.

**Declared In**

NSAttributedString.h

**attributesAtIndex:effectiveRange:**

Returns the attributes for the character at a given index.

```
- (NSDictionary *)attributesAtIndex:(NSUInteger)index
    effectiveRange:(NSRangePointer)aRange
```

**Parameters***index*

The index for which to return attributes. This value must lie within the bounds of the receiver.

*aRange*Upon return, the range over which the attributes and values are the same as those at *index*. This range isn't necessarily the maximum range covered, and its extent is implementation-dependent. If you need the maximum range, use[attributesAtIndex:longestEffectiveRange:inRange:](#) (page 10). If you don't need this value, pass NULL.**Return Value**The attributes for the character at *index*.**Discussion**Raises an `NSRangeException` if *index* lies beyond the end of the receiver's characters.**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [attributeAtIndex:effectiveRange:](#) (page 7)**Declared In**

NSAttributedString.h

**attributesAtIndex:longestEffectiveRange:inRange:**

Returns the attributes for the character at a given index, and by reference the range over which the attributes apply.

```
- (NSDictionary *)attributesAtIndex:(NSUInteger)index
    longestEffectiveRange:(NSRangePointer)aRange inRange:(NSRange)rangeLimit
```

**Parameters***index*

The index for which to return attributes. This value must not exceed the bounds of the receiver.

*aRange*If non-NULL, upon return contains the maximum range over which the attributes and values are the same as those at *index*, clipped to *rangeLimit*.

*rangeLimit*

The range over which to search for continuous presence of the attributes at *index*. This value must not exceed the bounds of the receiver.

**Discussion**

Raises an `NSRangeException` if *index* or any part of *rangeLimit* lies beyond the end of the receiver's characters.

If you don't need the range information, it's far more efficient to use the [attributesAtIndex:effectiveRange:](#) (page 10) method to retrieve the attribute value.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [attributeAtIndex:longestEffectiveRange:inRange:](#) (page 8)

**Declared In**

NSAttributedString.h

**initWithAttributedString:**

Returns an `NSAttributedString` object initialized with the characters and attributes of another given attributed string.

```
- (id)initWithAttributedString:(NSAttributedString *)attributedString
```

**Parameters**

*attributedString*

An attributed string.

**Return Value**

An `NSAttributedString` object initialized with the characters and attributes of *attributedString*. The returned object might be different than the original receiver.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [initWithRTF:documentAttributes:](#) (NSAttributedString Additions)

**Related Sample Code**

Sketch-112

**Declared In**

NSAttributedString.h

**initWithString:**

Returns an `NSAttributedString` object initialized with the characters of a given string and no attribute information.

```
- (id)initWithString:(NSString *)aString
```

### Parameters

*aString*

The characters for the new object.

### Return Value

An NSAttributedString object initialized with the characters of *aString* and no attribute information. The returned object might be different than the original receiver.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- initWithRTF:documentAttributes: (NSAttributedString Additions)

### Declared In

NSAttributedString.h

## initWithString:attributes:

Returns an NSAttributedString object initialized with a given string and attributes.

```
- (id)initWithString:(NSString *)aString attributes:(NSDictionary *)attributes
```

### Parameters

*aString*

The string for the new attributed string.

*attributes*

The attributes for the new attributed string. You can assign to a range of characters any attribute name/value pairs you wish, in addition to the standard attributes described in the “Constants” section of *NSAttributedString Application Kit Additions Reference*.

### Discussion

Returns an NSAttributedString object initialized with the characters of *aString* and the attributes of *attributes*. The returned object might be different from the original receiver.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- initWithRTF:documentAttributes: (NSAttributedString Additions)

### Related Sample Code

Aperture Edit Plugin - Borders & Titles

CIAnnotation

OpenGL Screensaver

### Declared In

NSAttributedString.h

## isEqualToAttributedString:

Returns a Boolean value that indicates whether the receiver is equal to another given attributed string.

- (BOOL)isEqualToString:(NSAttributedString \*)*otherString*

**Parameters**

*otherString*

The attributed string with which to compare the receiver.

**Return Value**

YES if the receiver is equal to *otherString*, otherwise NO.

**Discussion**

Attributed strings must match in both characters and attributes to be equal.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSAttributedString.h

## length

Returns the length of the receiver's string object.

- (NSUInteger)length

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

length (NSString)

- size (NSAttributedString Additions)

**Related Sample Code**

NumberInput\_IMKit\_Sample

VertexPerformanceTest

**Declared In**

NSAttributedString.h

## string

Returns the character contents of the receiver as an NSString object.

- (NSString \*)string

**Return Value**

The character contents of the receiver as an NSString object.

**Discussion**

This method doesn't strip out attachment characters; use NSText's string method to extract just the linguistically significant characters.

For performance reasons, this method returns the current backing store of the attributed string object. If you want to maintain a snapshot of this as you manipulate the returned string, you should make a copy of the appropriate substring.

This primitive method must guarantee efficient access to an attributed string's characters; subclasses should implement it to execute in  $O(1)$  time.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

iSpend

NumberInput\_IMKit\_Sample

Spotlight

**Declared In**

NSAttributedString.h

## Constants

Standard attribute keys are described in the "Constants" section of *NSAttributedString Application Kit Additions Reference*.

# Document Revision History

---

This table describes the changes to *NSAttributedString Class Reference*.

Date	Notes
2008-10-15	Added cautionary note about isEqual: comparison of NSAttributedString objects having attributes such as attachments, lists, and tables.
2006-06-28	Added a link to the string attribute constants.
	Added a link to the string attribute constants.
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History



# Index

---

## A

---

attribute:atIndex:effectiveRange: **instance method 7**  
attribute:atIndex:longestEffectiveRange:inRange: **instance method 8**  
attributedStringFromRange: **instance method 9**  
attributesAtIndex:effectiveRange: **instance method 10**  
attributesAtIndex:longestEffectiveRange:inRange: **instance method 10**

## I

---

initWithAttributedString: **instance method 11**  
initWithString: **instance method 11**  
initWithString:attributes: **instance method 12**  
isEqualToAttributedString: **instance method 12**

## L

---

length **instance method 13**

## S

---

string **instance method 13**