

---

# NSMutableAttributedString Class Reference

Data Management: Strings, Text, & Fonts



2010-02-25



Apple Inc.  
© 2010 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, iPhone, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

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

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

## **NSMutableAttributedString Class Reference 5**

---

Overview 5

Tasks 6

Retrieving Character Information 6

Changing Characters 6

Changing Attributes 6

Changing Characters and Attributes 6

Grouping Changes 7

Instance Methods 7

addAttribute:value:range: 7

addAttributes:range: 7

appendAttributedString: 8

beginEditing 8

deleteCharactersInRange: 9

endEditing 9

insertAttributedString:atIndex: 9

mutableString 10

removeAttribute:range: 10

replaceCharactersInRange:withAttributedString: 11

replaceCharactersInRange:withString: 11

setAttributedString: 12

setAttributes:range: 12

## **Document Revision History 15**

---



# NSMutableAttributedString Class Reference

---

<b>Inherits from</b>	NSAttributedString : NSObject
<b>Conforms to</b>	NSCoding (NSAttributedString) NSCopying (NSAttributedString) NSMutableCopying (NSAttributedString) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in iOS 3.2 and later.
<b>Companion guide</b>	Attributed String Programming Guide
<b>Declared in</b>	NSAttributedString.h

## Overview

`NSMutableAttributedString` declares the programmatic interface to objects that manage mutable attributed strings. You can add and remove characters (raw strings) and attributes separately or together as attributed strings. See the class description for `NSAttributedString` for more information about attributed strings.

When working with the Application Kit, you must also clean up changed attributes using the various `fix...` methods. See “Changing an Attributed String” for more information on fixing attributes. These methods, as well as others involving setting graphical attributes, are described in `NSMutableAttributedString` Additions in the Application Kit.

`NSMutableAttributedString` adds two primitive methods to those of `NSAttributedString`. These primitive methods provide the basis for all the other methods in its class. The primitive `replaceCharactersInRange:withString:` (page 11) method replaces a range of characters with those from a string, leaving all attribute information outside that range intact. The primitive `setAttributes:range:` (page 12) method sets attributes and values for a given range of characters, replacing any previous attributes and values for that range.

In Mac OS X, the Application Kit also uses `NSParagraphStyle` and its subclass `NSMutableParagraphStyle` to encapsulate the paragraph or ruler attributes used by the `NSAttributedString` classes.

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:`.

**iOS Note:** In iOS, this class is used primarily in conjunction with the Core Text framework.

## Tasks

### Retrieving Character Information

- [mutableString](#) (page 10)  
Returns the character contents of the receiver as an `NSMutableString` object.

### Changing Characters

- [replaceCharactersInRange:withString:](#) (page 11)  
Replaces the characters in the given range with the characters of the given string.
- [deleteCharactersInRange:](#) (page 9)  
Deletes the characters in the given range along with their associated attributes.

### Changing Attributes

- [setAttributes:range:](#) (page 12)  
Sets the attributes for the characters in the specified range to the specified attributes.
- [addAttribute:value:range:](#) (page 7)  
Adds an attribute with the given name and value to the characters in the specified range.
- [addAttributes:range:](#) (page 7)  
Adds the given collection of attributes to the characters in the specified range.
- [removeAttribute:range:](#) (page 10)  
Removes the named attribute from the characters in the specified range.

### Changing Characters and Attributes

- [appendAttributedString:](#) (page 8)  
Adds the characters and attributes of a given attributed string to the end of the receiver.
- [insertAttributedString:atIndex:](#) (page 9)  
Inserts the characters and attributes of the given attributed string into the receiver at the given index.
- [replaceCharactersInRange:withAttributedString:](#) (page 11)  
Replaces the characters and attributes in a given range with the characters and attributes of the given attributed string.
- [setAttributedString:](#) (page 12)  
Replaces the receiver's entire contents with the characters and attributes of the given attributed string.

## Grouping Changes

- [beginEditing](#) (page 8)  
Overridden by subclasses to buffer or optimize a series of changes to the receiver's characters or attributes, until it receives a matching [endEditing](#) (page 9) message, upon which it can consolidate changes and notify any observers that it has changed.
- [endEditing](#) (page 9)  
Overridden by subclasses to consolidate changes made since a previous [beginEditing](#) (page 8) message and to notify any observers of the changes.

## Instance Methods

### addAttribute:value:range:

Adds an attribute with the given name and value to the characters in the specified range.

```
- (void)addAttribute:(NSString *)name value:(id)value range:(NSRange)aRange
```

#### Parameters

*name*

A string specifying the attribute name. Attribute keys can be supplied by another framework or can be custom ones you define. For information about where to find the system-supplied attribute keys, see the overview section in *NSAttributedString Class Reference*.

*value*

The attribute value associated with *name*.

*aRange*

The range of characters to which the specified attribute/value pair applies.

#### Discussion

You may assign any *name/value* pair you wish to a range of characters, in addition to the standard attributes described in the “Constants” section of *NSAttributedString Additions*. Raises an `NSInvalidArgumentException` if *name* or *value* is `nil` and an `NSRangeException` if any part of *aRange* lies beyond the end of the receiver's characters.

#### Availability

Available in iOS 3.2 and later.

#### See Also

- [addAttributes:range:](#) (page 7)
- [removeAttribute:range:](#) (page 10)

#### Declared In

`NSAttributedString.h`

### addAttributes:range:

Adds the given collection of attributes to the characters in the specified range.

```
- (void)addAttributes:(NSDictionary *)attributes range:(NSRange)aRange
```

**Parameters***attributes*

A dictionary containing the attributes to add. Attribute keys can be supplied by another framework or can be custom ones you define. For information about where to find the system-supplied attribute keys, see the overview section in *NSAttributedString Class Reference*.

*aRange*

The range of characters to which the specified attributes apply.

**Discussion**

You may assign any name/value pair you wish to a range of characters, in addition to the standard attributes described in the “Constants” section of *NSAttributedString Additions*. Raises an `NSInvalidArgumentException` if *attributes* is `nil` and an `NSRangeException` if any part of *aRange* lies beyond the end of the receiver’s characters.

**Availability**

Available in iOS 3.2 and later.

**See Also**

- [addAttribute:value:range:](#) (page 7)
- [removeAttribute:range:](#) (page 10)

**Declared In**

`NSAttributedString.h`

**appendAttributedString:**

Adds the characters and attributes of a given attributed string to the end of the receiver.

```
- (void)appendAttributedString:(NSAttributedString *)attributedString
```

**Parameters***attributedString*

The string whose characters and attributes are added.

**Availability**

Available in iOS 3.2 and later.

**See Also**

- [insertAttributedString:atIndex:](#) (page 9)

**Declared In**

`NSAttributedString.h`

**beginEditing**

Overridden by subclasses to buffer or optimize a series of changes to the receiver’s characters or attributes, until it receives a matching [endEditing](#) (page 9) message, upon which it can consolidate changes and notify any observers that it has changed.

```
- (void)beginEditing
```

**Discussion**

You can nest pairs of `beginEditing` and `endEditing` (page 9) messages.



### Availability

Available in iOS 3.2 and later.

### Declared In

NSAttributedString.h

## deleteCharactersInRange:

Deletes the characters in the given range along with their associated attributes.

- (void)deleteCharactersInRange:(NSRange) *aRange*

### Parameters

*aRange*

A range specifying the characters to delete.

### Discussion

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

### Availability

Available in iOS 3.2 and later.

### See Also

- [replaceCharactersInRange:withAttributedString:](#) (page 11)

- [replaceCharactersInRange:withString:](#) (page 11)

### Declared In

NSAttributedString.h

## endEditing

Overridden by subclasses to consolidate changes made since a previous [beginEditing](#) (page 8) message and to notify any observers of the changes.

- (void)endEditing

### Discussion

The `NSMutableAttributedString` implementation does nothing. `NSTextStorage`, for example, overrides this method to invoke `fixAttributesInRange:` and to inform its `NSLayoutManager` objects that they need to re-lay the text.

### Availability

Available in iOS 3.2 and later.

### Declared In

NSAttributedString.h

## insertAttributedString:atIndex:

Inserts the characters and attributes of the given attributed string into the receiver at the given index.

```
- (void)insertAttributedString:(NSAttributedString *)attributedString
  atIndex:(NSUInteger)index
```

**Parameters**

*attributedString*

The string whose characters and attributes are inserted.

*index*

The index at which the characters and attributes are inserted.

**Discussion**

The new characters and attributes begin at the given index and the existing characters and attributes from the index to the end of the receiver are shifted by the length of the attributed string. Raises an `NSRangeException` if *index* lies beyond the end of the receiver's characters.

**Availability**

Available in iOS 3.2 and later.

**See Also**

- [appendAttributedString:](#) (page 8)

**Declared In**

NSAttributedString.h

## mutableString

Returns the character contents of the receiver as an `NSMutableString` object.

```
- (NSMutableString *)mutableString
```

**Return Value**

The mutable string object.

**Discussion**

The receiver tracks changes to this string and keeps its attribute mappings up to date.

**Availability**

Available in iOS 3.2 and later.

**Declared In**

NSAttributedString.h

## removeAttribute:range:

Removes the named attribute from the characters in the specified range.

```
- (void)removeAttribute:(NSString *)name range:(NSRange)aRange
```

**Parameters**

*name*

A string specifying the attribute name to remove. Attribute keys can be supplied by another framework or can be custom ones you define. For information about where to find the system-supplied attribute keys, see the overview section in *NSAttributedString Class Reference*.

*aRange*

The range of characters from which the specified attribute is removed.

#### Discussion

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

#### Availability

Available in iOS 3.2 and later.

#### See Also

- [addAttribute:value:range:](#) (page 7)
- [addAttributes:range:](#) (page 7)

#### Declared In

NSAttributedString.h

## replaceCharactersInRange:withAttributedString:

Replaces the characters and attributes in a given range with the characters and attributes of the given attributed string.

```
- (void)replaceCharactersInRange:(NSRange)aRange
    withAttributedString:(NSAttributedString *)attributedString
```

#### Parameters

*aRange*

The range of characters and attributes replaced.

*attributedString*

The attributed string whose characters and attributes replace those in the specified range.

#### Discussion

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

#### Availability

Available in iOS 3.2 and later.

#### See Also

- [insertAttributedString:atIndex:](#) (page 9)

#### Declared In

NSAttributedString.h

## replaceCharactersInRange:withString:

Replaces the characters in the given range with the characters of the given string.

```
- (void)replaceCharactersInRange:(NSRange)aRange withString:(NSString *)aString
```

#### Parameters

*aRange*

A range specifying the characters to replace.

*aString*

A string specifying the characters to replace those in *aRange*.

#### Discussion

The new characters inherit the attributes of the first replaced character from *aRange*. Where the length of *aRange* is 0, the new characters inherit the attributes of the character preceding *aRange* if it has any, otherwise of the character following *aRange*.

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

#### Availability

Available in iOS 3.2 and later.

#### See Also

- [deleteCharactersInRange:](#) (page 9)

#### Declared In

NSAttributedString.h

## setAttributedString:

Replaces the receiver's entire contents with the characters and attributes of the given attributed string.

```
- (void)setAttributedString:(NSAttributedString *)attributedString
```

#### Parameters

*attributedString*

The attributed string whose characters and attributes replace those in the receiver.

#### Availability

Available in iOS 3.2 and later.

#### See Also

- [appendAttributedString:](#) (page 8)

#### Declared In

NSAttributedString.h

## setAttributes:range:

Sets the attributes for the characters in the specified range to the specified attributes.

```
- (void)setAttributes:(NSDictionary *)attributes range:(NSRange)aRange
```

#### Parameters

*attributes*

A dictionary containing the attributes to set. Attribute keys can be supplied by another framework or can be custom ones you define. For information about where to find the system-supplied attribute keys, see the overview section in *NSAttributedString Class Reference*.

*aRange*

The range of characters whose attributes are set.

### Discussion

These new attributes replace any attributes previously associated with the characters in *aRange*. Raises an `NSRangeException` if any part of *aRange* lies beyond the end of the receiver's characters.

To set attributes for a zero-length `NSMutableAttributedString` displayed in a text view, use the `NSTextView` method `setTypingAttributes:`.

### Availability

Available in iOS 3.2 and later.

### See Also

- [addAttributes:range:](#) (page 7)
- [removeAttribute:range:](#) (page 10)

### Declared In

`NSAttributedString.h`



# Document Revision History

---

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

Date	Notes
2010-02-25	Added this class to the Foundation framework in iOS 3.2.
2006-06-28	Fixed a bad link to "Changing an Attributed String."
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History