
NSMutableString Class Reference

[Cocoa > Data Management](#)



2007-03-06



Apple Inc.
© 2007 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.

iPhone is a trademark of Apple 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

NSString Class Reference 5

Overview 5

Tasks 6

 Creating and Initializing a Mutable String 6

 Modifying a String 6

Class Methods 6

 stringWithCapacity: 6

Instance Methods 7

 appendFormat: 7

 appendString: 8

 deleteCharactersInRange: 8

 initWithCapacity: 9

 insertString:atIndex: 9

 replaceCharactersInRange:withString: 10

 replaceOccurrencesOfString:withString:options:range: 10

 setString: 11

Document Revision History 13

Index 15

NSMutableString Class Reference

Inherits from	NSString : NSObject
Conforms to	NSCoding (NSString) NSCopying (NSString) NSMutableCopying (NSString) NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	String Programming Guide for Cocoa
Declared in	NSString.h
Related sample code	CoreRecipes CubePuzzle NSGLImage NumberInput_IMKit_Sample QTAudioExtractionPanel

Overview

The `NSMutableString` class declares the programmatic interface to an object that manages a mutable string—that is, a string whose contents can be edited—that conceptually represents an array of Unicode characters. To construct and manage an immutable string—or a string that cannot be changed after it has been created—use an object of the `NSString` class.

The `NSMutableString` class adds one primitive method—[replaceCharactersInRange:withString:](#) (page 10)—to the basic string-handling behavior inherited from `NSString`. All other methods that modify a string work through this method. For example, [insertString:atIndex:](#) (page 9) simply replaces the characters in a range of 0 length, while [deleteCharactersInRange:](#) (page 8) replaces the characters in a given range with no characters.

Tasks

Creating and Initializing a Mutable String

- + [stringWithCapacity:](#) (page 6)
Returns an empty NSMutableString object with initial storage for a given number of characters.
- [initWithCapacity:](#) (page 9)
Returns an NSMutableString object initialized with initial storage for a given number of characters,

Modifying a String

- [appendFormat:](#) (page 7)
Adds a constructed string to the receiver.
- [appendString:](#) (page 8)
Adds to the end of the receiver the characters of a given string.
- [deleteCharactersInRange:](#) (page 8)
Removes from the receiver the characters in a given range.
- [insertString:atIndex:](#) (page 9)
Inserts into the receiver the characters of a given string at a given location.
- [replaceCharactersInRange:withString:](#) (page 10)
Replaces the characters from *aRange* with those in *aString*.
- [replaceOccurrencesOfString:withString:options:range:](#) (page 10)
Replaces all occurrences of a given string in a given range with another given string, returning the number of replacements.
- [setString:](#) (page 11)
Replaces the characters of the receiver with those in a given string.

Class Methods

stringWithCapacity:

Returns an empty NSMutableString object with initial storage for a given number of characters.

```
+ (id)stringWithCapacity:(NSUInteger)capacity
```

Parameters

capacity

The number of characters the string is expected to initially contain.

Return Value

An empty NSMutableString object with initial storage for *capacity* characters.

Discussion

The number of characters indicated by *capacity* is simply a hint to increase the efficiency of data storage. The value does *not* limit the length of the string.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

PDFKitLinker2

QTRecorder

Declared In

NSString.h

Instance Methods

appendFormat:

Adds a constructed string to the receiver.

```
- (void)appendFormat:(NSString *)format ...
```

Parameters

format

A format string. See Formatting String Objects for more information. This value must not be `nil`.

Important: Raises an `NSInvalidArgumentException` if *format* is `nil`.

...

A comma-separated list of arguments to substitute into *format*.

Discussion

The appended string is formed using `NSString`'s `stringWithFormat:` method with the arguments listed.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Related Sample Code

CoreRecipes

iSpend

ThreadsImporter

ThreadsImportMovie

TimelineToTC

Declared In

NSString.h

appendString:

Adds to the end of the receiver the characters of a given string.

- (void)appendString:(NSString *)*aString*

Parameters

aString

The string to append to the receiver. *aString* must not be nil

Availability

Available in Mac OS X v10.0 and later.

See Also

- [appendFormat:](#) (page 7)

Related Sample Code

CoreRecipes

DockTile

JavaSplashScreen

SampleScannerApp

TimelineToTC

Declared In

NSString.h

deleteCharactersInRange:

Removes from the receiver the characters in a given range.

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

Parameters

aRange

The range of characters to delete. *aRange* must not exceed the bounds of the receiver.

Important: Raises an `NSRangeException` if any part of *aRange* lies beyond the end of the string.

Discussion

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.

Related Sample Code

NumberInput_IMKit_Sample

Declared In

NSString.h

initWithCapacity:

Returns an `NSMutableString` object initialized with initial storage for a given number of characters,

```
- (id)initWithCapacity:(NSUInteger)capacity
```

Parameters

capacity

The number of characters the string is expected to initially contain.

Return Value

An initialized `NSMutableString` object with initial storage for *capacity* characters. The returned object might be different than the original receiver.

Discussion

The number of characters indicated by *capacity* is simply a hint to increase the efficiency of data storage. The value does *not* limit the length of the string.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSString.h`

insertString:atIndex:

Inserts into the receiver the characters of a given string at a given location.

```
- (void)insertString:(NSString *)aString atIndex:(NSUInteger)anIndex
```

Parameters

aString

The string to insert into the receiver. *aString* must not be `nil`.

anIndex

The location at which *aString* is inserted. The location must not exceed the bounds of the receiver.

Important: Raises an `NSRangeException` if *anIndex* lies beyond the end of the string.

Discussion

The new characters begin at *anIndex* and the existing characters from *anIndex* to the end are shifted by the length of *aString*.

This method treats the length of the string as a valid index value that returns an empty string.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Aperture Image Resizer

CustomSave

Declared In

`NSString.h`

replaceCharactersInRange:withString:

Replaces the characters from *aRange* with those in *aString*.

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

Parameters

aRange

The range of characters to replace. *aRange* must not exceed the bounds of the receiver.

Important: Raises an `NSRangeException` if any part of *aRange* lies beyond the end of the receiver.

aString

The string with which to replace the characters in *aRange*. *aString* must not be `nil`.

Discussion

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.

Related Sample Code

LSMSmartCategorizer

Declared In

NSString.h

replaceOccurrencesOfString:withString:options:range:

Replaces all occurrences of a given string in a given range with another given string, returning the number of replacements.

```
- (NSUInteger)replaceOccurrencesOfString:(NSString *)target withString:(NSString *)replacement options:(NSStringCompareOptions)opts range:(NSRange)searchRange
```

Parameters*target*

The string to replace.

Important: Raises an `NSInvalidArgumentException` if *target* is `nil`.

*replacement*The string with which to replace *target*.

Important: Raises an `NSInvalidArgumentException` if *replacement* is `nil`.

*opts*A mask specifying search options. See *String Programming Guide for Cocoa* for details.

If *opts* is `NSBackwardsSearch`, the search is done from the end of the range. If *opts* is `NSAnchoredSearch`, only anchored (but potentially multiple) instances are replaced. `NSLiteralSearch` and `NSCaseInsensitiveSearch` also apply.

searchRange

The range of characters to replace. *aRange* must not exceed the bounds of the receiver. Specify *searchRange* as `NSMakeRange(0, [receiver length])` to process the entire string.

Important: Raises an `NSRangeException` if any part of *searchRange* lies beyond the end of the receiver.

Return Value

The number of replacements made.

Discussion

This method treats the length of the string as a valid range value that returns an empty string.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

CubePuzzle

JavaSplashScreen

NewsReader

SpotlightAPI

Video Hardware Info

Declared In

NSString.h

setString:

Replaces the characters of the receiver with those in a given string.

```
- (void)setString:(NSString *)aString
```

Parameters

aString

The string with which to replace the receiver's content. *aString* must not be `nil`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

NumberInput_IMKit_Sample

PDFKitLinker2

QTCoreVideo201

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

NSString.h

Document Revision History

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

Date	Notes
2007-03-06	Updated for Mac OS X v10.5.
2006-05-23	Corrected minor typographical errors.
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

appendFormat: **instance method** [7](#)
appendString: **instance method** [8](#)

D

deleteCharactersInRange: **instance method** [8](#)

I

initWithCapacity: **instance method** [9](#)
insertString:atIndex: **instance method** [9](#)

R

replaceCharactersInRange:withString: **instance method** [10](#)
replaceOccurrencesOfString:withString:options:range: **instance method** [10](#)

S

setString: **instance method** [11](#)
stringWithCapacity: **class method** [6](#)