

---

# NSString 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, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

Numbers 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
Adopted Protocols	6
Tasks	6
Creating a Standard Character Set	6
Creating a Custom Character Set	7
Creating and Managing Character Sets as Bitmap Representations	7
Testing Set Membership	8
Class Methods	8
alphanumericCharacterSet	8
capitalizedLetterCharacterSet	8
characterSetWithBitmapRepresentation:	9
characterSetWithCharactersInString:	9
characterSetWithContentsOfFile:	10
characterSetWithRange:	10
controlCharacterSet	11
decimalDigitCharacterSet	11
decomposableCharacterSet	12
illegalCharacterSet	12
letterCharacterSet	13
lowercaseLetterCharacterSet	13
newlineCharacterSet	14
nonBaseCharacterSet	14
punctuationCharacterSet	14
symbolCharacterSet	15
uppercaseLetterCharacterSet	15
whitespaceAndNewlineCharacterSet	16
whitespaceCharacterSet	16
Instance Methods	17
bitmapRepresentation	17
characterIsMember:	17
hasMemberInPlane:	18
invertedSet	18
isSupersetOfSet:	18
longCharacterIsMember:	19
Constants	19
NSOpenStepUnicodeReservedBase	19

**Document Revision History 21**

---

**Index 23**

---

# NSString 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>	String Programming Guide for Cocoa
<b>Declared in</b>	NSString.h
<b>Related sample code</b>	ImageClient iSpend Quartz Composer WWDC 2005 TextEdit TextEditPlus VertexPerformanceTest

## Overview

An `NSString` object represents a set of Unicode-compliant characters. `NSString` and `NSStringScanner` objects use `NSString` objects to group characters together for searching operations, so that they can find any of a particular set of characters during a search. The cluster's two public classes, `NSString` and `NSMutableString`, declare the programmatic interface for static and dynamic character sets, respectively.

The objects you create using these classes are referred to as character set objects (and when no confusion will result, merely as character sets). Because of the nature of class clusters, character set objects aren't actual instances of the `NSString` or `NSMutableString` classes but of one of their private subclasses. Although a character set object's class is private, its interface is public, as declared by these abstract superclasses, `NSString` and `NSMutableString`. The character set classes adopt the `NSCopying` and `NSMutableCopying` protocols, making it convenient to convert a character set of one type to the other.

The `NSString` class declares the programmatic interface for an object that manages a set of Unicode characters (see the `NSString` class cluster specification for information on Unicode). `NSString`'s principal primitive method, `characterIsMember:` (page 17), provides the basis for all other instance methods in its interface. A subclass of `NSString` needs only to implement this method, plus

`mutableCopyWithZone:`, for proper behavior. For optimal performance, a subclass should also override `bitmapRepresentation` (page 17), which otherwise works by invoking `characterIsMember:` (page 17) for every possible Unicode value.

`NSCharacterSet` is “toll-free bridged” with its Cocoa Foundation counterpart, *CFCharacterSet Reference*. This means that the Core Foundation type is interchangeable in function or method calls with the bridged Foundation object. Therefore, in a method where you see an `NSCharacterSet *` parameter, you can pass a `CFCharacterSetRef`, and in a function where you see a `CFCharacterSetRef` parameter, you can pass an `NSCharacterSet` instance (you cast one type to the other to suppress compiler warnings). See *Interchangeable Data Types* for more information on toll-free bridging.

The mutable subclass of `NSCharacterSet` is `NSMutableCharacterSet`.

## Adopted Protocols

### NSCoding

`encodeWithCoder:`  
`initWithCoder:`

### NSCopying

`copyWithZone:`

### NSMutableCopying

`mutableCopyWithZone:`

## Tasks

### Creating a Standard Character Set

- + [alphanumericCharacterSet](#) (page 8)  
Returns a character set containing the characters in the categories Letters, Marks, and Numbers.
- + [capitalizedLetterCharacterSet](#) (page 8)  
Returns a character set containing the characters in the category of Titlecase Letters.
- + [controlCharacterSet](#) (page 11)  
Returns a character set containing the characters in the categories of Control or Format Characters.
- + [decimalDigitCharacterSet](#) (page 11)  
Returns a character set containing the characters in the category of Decimal Numbers.
- + [decomposableCharacterSet](#) (page 12)  
Returns a character set containing all individual Unicode characters that can also be represented as composed character sequences.
- + [illegalCharacterSet](#) (page 12)  
Returns a character set containing values in the category of Non-Characters or that have not yet been defined in version 3.2 of the Unicode standard.

- + [letterCharacterSet](#) (page 13)  
Returns a character set containing the characters in the categories Letters and Marks.
- + [lowercaseLetterCharacterSet](#) (page 13)  
Returns a character set containing the characters in the category of Lowercase Letters.
- + [newlineCharacterSet](#) (page 14)  
Returns a character set containing the newline characters.
- + [nonBaseCharacterSet](#) (page 14)  
Returns a character set containing the characters in the category of Marks.
- + [punctuationCharacterSet](#) (page 14)  
Returns a character set containing the characters in the category of Punctuation.
- + [symbolCharacterSet](#) (page 15)  
Returns a character set containing the characters in the category of Symbols.
- + [uppercaseLetterCharacterSet](#) (page 15)  
Returns a character set containing the characters in the categories of Uppercase Letters and Titlecase Letters.
- + [whitespaceAndNewlineCharacterSet](#) (page 16)  
Returns a character set containing only the whitespace characters space (U+0020) and tab (U+0009) and the newline and nextline characters (U+000A–U+000D, U+0085).
- + [whitespaceCharacterSet](#) (page 16)  
Returns a character set containing only the in-line whitespace characters space (U+0020) and tab (U+0009).

## Creating a Custom Character Set

- + [characterSetWithCharactersInString:](#) (page 9)  
Returns a character set containing the characters in a given string.
- + [characterSetWithRange:](#) (page 10)  
Returns a character set containing characters with Unicode values in a given range.
- [invertedSet](#) (page 18)  
Returns a character set containing only characters that don't exist in the receiver.

## Creating and Managing Character Sets as Bitmap Representations

- + [characterSetWithBitmapRepresentation:](#) (page 9)  
Returns a character set containing characters determined by a given bitmap representation.
- + [characterSetWithContentsOfFile:](#) (page 10)  
Returns a character set read from the bitmap representation stored in the file a given path.
- [bitmapRepresentation](#) (page 17)  
Returns an NSData object encoding the receiver in binary format.

## Testing Set Membership

- [characterIsMember:](#) (page 17)  
Returns a Boolean value that indicates whether a given character is in the receiver.
- [hasMemberInPlane:](#) (page 18)  
Returns a Boolean value that indicates whether the receiver has at least one member in a given character plane.
- [isSupersetOfSet:](#) (page 18)  
Returns a Boolean value that indicates whether the receiver is a superset of another given character set.
- [longCharacterIsMember:](#) (page 19)  
Returns a Boolean value that indicates whether a given long character is a member of the receiver.

## Class Methods

### alphanumericCharacterSet

Returns a character set containing the characters in the categories Letters, Marks, and Numbers.

```
+ (id)alphanumericCharacterSet
```

#### Return Value

A character set containing the characters in the categories Letters, Marks, and Numbers.

#### Discussion

Informally, this set is the set of all characters used as basic units of alphabets, syllabaries, ideographs, and digits.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- + [letterCharacterSet](#) (page 13)
- + [decimalDigitCharacterSet](#) (page 11)

#### Declared In

NSCharacterSet.h

### capitalizedLetterCharacterSet

Returns a character set containing the characters in the category of Titlecase Letters.

```
+ (id)capitalizedLetterCharacterSet
```

#### Return Value

A character set containing the characters in the category of Titlecase Letters.

#### Availability

Available in Mac OS X v10.2 and later.



**See Also**+ [LetterCharacterSet](#) (page 13)+ [uppercaseLetterCharacterSet](#) (page 15)**Declared In**

NSCharacterSet.h

**characterSetWithBitmapRepresentation:**

Returns a character set containing characters determined by a given bitmap representation.

+ (id)characterSetWithBitmapRepresentation:(NSData \*)*data***Parameters***data*

A bitmap representation of a character set.

**Return Value**A character set containing characters determined by *data*.**Discussion**

This method is useful for creating a character set object with data from a file or other external data source.

A raw bitmap representation of a character set is a byte array of  $2^{16}$  bits (that is, 8192 bytes). The value of the bit at position *n* represents the presence in the character set of the character with decimal Unicode value *n*. To add a character with decimal Unicode value *n* to a raw bitmap representation, use a statement such as the following:

```
unsigned char bitmapRep[8192];
bitmapRep[n >> 3] |= (((unsigned int)1) << (n & 7));
```

To remove that character:

```
bitmapRep[n >> 3] &= ~(((unsigned int)1) << (n & 7));
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [bitmapRepresentation](#) (page 17)+ [characterSetWithContentsOfFile:](#) (page 10)**Declared In**

NSCharacterSet.h

**characterSetWithCharactersInString:**

Returns a character set containing the characters in a given string.

+ (id)characterSetWithCharactersInString:(NSString \*)*aString*

**Parameters***aString*

A string containing characters for the new character set.

**Return Value**A character set containing the characters in *aString*. Returns an empty character set if *aString* is empty.**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

iSpend

QTAudioExtractionPanel

Quartz Composer QCTV

**Declared In**

NSCharacterSet.h

**characterSetWithContentsOfFile:**

Returns a character set read from the bitmap representation stored in the file a given path.

+ (id)characterSetWithContentsOfFile:(NSString \*)*path***Parameters***path*A path to a file containing a bitmap representation of a character set. The path name must end with the extension `.bitmap`.**Return Value**A character set read from the bitmap representation stored in the file at *path*.**Discussion**To read a bitmap representation from any file, use the `NSData` method `dataWithContentsOfFile:options:error:` and pass the result to [characterSetWithBitmapRepresentation:](#) (page 9).

This method doesn't use filenames to check for the uniqueness of the character sets it creates. To prevent duplication of character sets in memory, cache them and make them available through an API that checks whether the requested set has already been loaded.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSCharacterSet.h

**characterSetWithRange:**

Returns a character set containing characters with Unicode values in a given range.

+ (id)characterSetWithRange:(NSRange) *aRange*

**Parameters***aRange*

A range of Unicode values.

*aRange.location* is the value of the first character to return; *aRange.location* + *aRange.length* - 1 is the value of the last.**Return Value**A character set containing characters whose Unicode values are given by *aRange*. If *aRange.length* is 0, returns an empty character set.**Discussion**

This code excerpt creates a character set object containing the lowercase English alphabetic characters:

```

NSRange lcEnglishRange;
NSCharacterSet *lcEnglishLetters;

lcEnglishRange.location = (unsigned int)'a';
lcEnglishRange.length = 26;
lcEnglishLetters = [NSCharacterSet characterSetWithRange:lcEnglishRange];

```

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSCharacterSet.h

**controlCharacterSet**

Returns a character set containing the characters in the categories of Control or Format Characters.

+ (id)controlCharacterSet

**Return Value**

A character set containing the characters in the categories of Control or Format Characters.

**Discussion**

These characters are specifically the Unicode values U+0000 to U+001F and U+007F to U+009F.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**+ [illegalCharacterSet](#) (page 12)**Related Sample Code**

Link Snoop

**Declared In**

NSCharacterSet.h

**decimalDigitCharacterSet**

Returns a character set containing the characters in the category of Decimal Numbers.

+ (id)decimalDigitCharacterSet

#### Return Value

A character set containing the characters in the category of Decimal Numbers.

#### Discussion

Informally, this set is the set of all characters used to represent the decimal values 0 through 9. These characters include, for example, the decimal digits of the Indic scripts and Arabic.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [alphanumericCharacterSet](#) (page 8)

#### Declared In

NSCharacterSet.h

## decomposableCharacterSet

Returns a character set containing all individual Unicode characters that can also be represented as composed character sequences.

+ (id)decomposableCharacterSet

#### Return Value

A character set containing all individual Unicode characters that can also be represented as composed character sequences (such as for letters with accents), by the definition of “standard decomposition” in version 3.2 of the Unicode character encoding standard.

#### Discussion

These characters include compatibility characters as well as pre-composed characters.

**Note:** This character set doesn't currently include the Hangul characters defined in version 2.0 of the Unicode standard.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [nonBaseCharacterSet](#) (page 14)

#### Declared In

NSCharacterSet.h

## illegalCharacterSet

Returns a character set containing values in the category of Non-Characters or that have not yet been defined in version 3.2 of the Unicode standard.

+ (id)illegalCharacterSet

**Return Value**

A character set containing values in the category of Non-Characters or that have not yet been defined in version 3.2 of the Unicode standard.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [controlCharacterSet](#) (page 11)

**Declared In**

NSCharacterSet.h

## letterCharacterSet

Returns a character set containing the characters in the categories Letters and Marks.

```
+ (id)letterCharacterSet
```

**Return Value**

A character set containing the characters in the categories Letters and Marks.

**Discussion**

Informally, this set is the set of all characters used as letters of alphabets and ideographs.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [alphanumericCharacterSet](#) (page 8)

+ [lowercaseLetterCharacterSet](#) (page 13)

+ [uppercaseLetterCharacterSet](#) (page 15)

**Declared In**

NSCharacterSet.h

## lowercaseLetterCharacterSet

Returns a character set containing the characters in the category of Lowercase Letters.

```
+ (id)lowercaseLetterCharacterSet
```

**Return Value**

A character set containing the characters in the category of Lowercase Letters.

**Discussion**

Informally, this set is the set of all characters used as lowercase letters in alphabets that make case distinctions.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [uppercaseLetterCharacterSet](#) (page 15)

+ [letterCharacterSet](#) (page 13)

**Declared In**

NSCharacterSet.h

## newlineCharacterSet

Returns a character set containing the newline characters.

+ (id)newlineCharacterSet

**Return Value**

A character set containing the newline characters (U+000A–U+000D, U+0085).

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

+ [whitespaceAndNewlineCharacterSet](#) (page 16)

+ [whitespaceCharacterSet](#) (page 16)

**Declared In**

NSCharacterSet.h

## nonBaseCharacterSet

Returns a character set containing the characters in the category of Marks.

+ (id)nonBaseCharacterSet

**Return Value**

A character set containing the characters in the category of Marks.

**Discussion**

This set is also defined as all legal Unicode characters with a non-spacing priority greater than 0. Informally, this set is the set of all characters used as modifiers of base characters.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [decomposableCharacterSet](#) (page 12)

**Declared In**

NSCharacterSet.h

## punctuationCharacterSet

Returns a character set containing the characters in the category of Punctuation.

+ (id)punctuationCharacterSet

**Return Value**

A character set containing the characters in the category of Punctuation.

**Discussion**

Informally, this set is the set of all non-whitespace characters used to separate linguistic units in scripts, such as periods, dashes, parentheses, and so on.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSCharacterSet.h

**symbolCharacterSet**

Returns a character set containing the characters in the category of Symbols.

```
+ (id)symbolCharacterSet
```

**Return Value**

A character set containing the characters in the category of Symbols.

**Discussion**

These characters include, for example, the dollar sign (\$) and the plus (+) sign.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSCharacterSet.h

**uppercaseLetterCharacterSet**

Returns a character set containing the characters in the categories of Uppercase Letters and Titlecase Letters.

```
+ (id)uppercaseLetterCharacterSet
```

**Return Value**

A character set containing the characters in the categories of Uppercase Letters and Titlecase Letters.

**Discussion**

Informally, this set is the set of all characters used as uppercase letters in alphabets that make case distinctions.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [capitalizedLetterCharacterSet](#) (page 8)

+ [lowercaseLetterCharacterSet](#) (page 13)

+ [letterCharacterSet](#) (page 13)

**Declared In**

NSCharacterSet.h

## whitespaceAndNewlineCharacterSet

Returns a character set containing only the whitespace characters space (U+0020) and tab (U+0009) and the newline and nextline characters (U+000A–U+000D, U+0085).

```
+ (id)whitespaceAndNewlineCharacterSet
```

### Return Value

A character set containing only the whitespace characters space (U+0020) and tab (U+0009) and the newline and nextline characters (U+000A–U+000D, U+0085).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [newlineCharacterSet](#) (page 14)

+ [whitespaceCharacterSet](#) (page 16)

### Related Sample Code

ImageMapExample

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

TextLinks

VertexPerformanceTest

### Declared In

NSCharacterSet.h

## whitespaceCharacterSet

Returns a character set containing only the in-line whitespace characters space (U+0020) and tab (U+0009).

```
+ (id)whitespaceCharacterSet
```

### Return Value

A character set containing only the in-line whitespace characters space (U+0020) and tab (U+0009).

### Discussion

This set doesn't contain the newline or carriage return characters.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [whitespaceAndNewlineCharacterSet](#) (page 16)

+ [newlineCharacterSet](#) (page 14)

### Related Sample Code

CoreRecipes

ImageClient

### Declared In

NSCharacterSet.h



## Instance Methods

### bitmapRepresentation

Returns an `NSData` object encoding the receiver in binary format.

```
- (NSData *)bitmapRepresentation
```

#### Return Value

An `NSData` object encoding the receiver in binary format.

#### Discussion

This format is suitable for saving to a file or otherwise transmitting or archiving.

A raw bitmap representation of a character set is a byte array of  $2^{16}$  bits (that is, 8192 bytes). The value of the bit at position  $n$  represents the presence in the character set of the character with decimal Unicode value  $n$ . To test for the presence of a character with decimal Unicode value  $n$  in a raw bitmap representation, use an expression such as the following:

```
unsigned char bitmapRep[8192];
if (bitmapRep[n >> 3] & (((unsigned int)1) << (n & 7))) {
    /* Character is present. */
}
```

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [characterSetWithBitmapRepresentation:](#) (page 9)

#### Declared In

`NSCharacterSet.h`

### characterIsMember:

Returns a Boolean value that indicates whether a given character is in the receiver.

```
- (BOOL)characterIsMember:(unichar)aCharacter
```

#### Parameters

*aCharacter*

The character to test for membership of the receiver.

#### Return Value

YES if *aCharacter* is in the receiving character set, otherwise NO.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [longCharacterIsMember:](#) (page 19)

**Declared In**

NSCharacterSet.h

**hasMemberInPlane:**

Returns a Boolean value that indicates whether the receiver has at least one member in a given character plane.

```
- (BOOL)hasMemberInPlane:(uint8_t)thePlane
```

**Parameters**

*thePlane*

A character plane.

**Return Value**

YES if the receiver has at least one member in *thePlane*, otherwise NO.

**Discussion**

This method makes it easier to find the plane containing the members of the current character set. The Basic Multilingual Plane is plane 0.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

NSCharacterSet.h

**invertedSet**

Returns a character set containing only characters that don't exist in the receiver.

```
- (NSCharacterSet *)invertedSet
```

**Return Value**

A character set containing only characters that don't exist in the receiver.

**Discussion**

Inverting an immutable character set is much more efficient than inverting a mutable character set.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

`invert (NSMutableCharacterSet)`

**Declared In**

NSCharacterSet.h

**isSupersetOfSet:**

Returns a Boolean value that indicates whether the receiver is a superset of another given character set.

```
- (BOOL)isSupersetOfSet:(NSCharacterSet *)theOtherSet
```

**Parameters***theOtherSet*

A character set.

**Return Value**YES if the receiver is a superset of *theOtherSet*, otherwise NO.**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

NSCharacterSet.h

**longCharacterIsMember:**

Returns a Boolean value that indicates whether a given long character is a member of the receiver.

- (BOOL)longCharacterIsMember:(UTF32Char) *theLongChar***Parameters***theLongChar*

A UTF32 character.

**Return Value**YES if *theLongChar* is in the receiver, otherwise NO.**Discussion**

This method supports the specification of 32-bit characters.

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**- [characterIsMember:](#) (page 17)**Declared In**

NSCharacterSet.h

## Constants

**NSOpenStepUnicodeReservedBase**

Specifies lower bound for a Unicode character range reserved for Apple's corporate use.

```
enum {  
    NSOpenStepUnicodeReservedBase = 0xF400  
};
```

**Constants**

NSOpenStepUnicodeReservedBase

Specifies lower bound for a Unicode character range reserved for Apple's corporate use (the range is 0xF400-0xF8FF).

Available in Mac OS X v10.0 and later.

Declared in NSCharacterSet.h.

**Declared In**

NSCharacterSet.h

# Document Revision History

---

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

Date	Notes
2008-10-15	Corrected typographical error.
2007-02-22	Updated to include API introduced in Mac OS X v10.5. Revised task groupings.
2006-05-23	First publication of this content as a separate document. Removed Unicode version number.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

alphanumericCharacterSet **class method** [8](#)

## B

---

bitmapRepresentation **instance method** [17](#)

## C

---

capitalizedLetterCharacterSet **class method** [8](#)

characterIsMember: **instance method** [17](#)

characterSetWithBitmapRepresentation: **class method** [9](#)

characterSetWithCharactersInString: **class method** [9](#)

characterSetWithContentsOfFile: **class method** [10](#)

characterSetWithRange: **class method** [10](#)

controlCharacterSet **class method** [11](#)

## D

---

decimalDigitCharacterSet **class method** [11](#)

decomposableCharacterSet **class method** [12](#)

## H

---

hasMemberInPlane: **instance method** [18](#)

## I

---

illegalCharacterSet **class method** [12](#)

invertedSet **instance method** [18](#)

isSupersetOfSet: **instance method** [18](#)

## L

---

letterCharacterSet **class method** [13](#)

longCharacterIsMember: **instance method** [19](#)

lowercaseLetterCharacterSet **class method** [13](#)

## N

---

newlineCharacterSet **class method** [14](#)

nonBaseCharacterSet **class method** [14](#)

NSOpenStepUnicodeReservedBase [19](#)

NSOpenStepUnicodeReservedBase **constant** [20](#)

## P

---

punctuationCharacterSet **class method** [14](#)

## S

---

symbolCharacterSet **class method** [15](#)

## U

---

uppercaseLetterCharacterSet **class method** [15](#)

## W

---

whitespaceAndNewlineCharacterSet **class method** [16](#)

whitespaceCharacterSet **class method** [16](#)