
NSDateFormatter Class Reference

[Cocoa](#) > [User Experience](#)



2008-11-19



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.

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

NSDateFormatter Class Reference 7

Overview	7
Tasks	8
Initializing a Date Formatter	8
Managing Behavior	8
Converting Objects	9
Managing Formats and Styles	9
Managing Attributes	9
Managing AM and PM Symbols	10
Managing Weekday Symbols	10
Managing Month Symbols	11
Managing Quarter Symbols	11
Managing Era Symbols	12
Class Methods	12
defaultFormatterBehavior	12
setDefaultFormatterBehavior:	13
Instance Methods	13
allowsNaturalLanguage	13
AMSymbol	14
calendar	14
dateFormat	14
dateFromString:	15
dateStyle	15
defaultDate	16
eraSymbols	16
formatterBehavior	16
generatesCalendarDates	17
getObjectValue:forString:range:error:	17
gregorianStartDate	18
init	18
initWithDateFormat:allowNaturalLanguage:	19
isLenient	20
locale	20
longEraSymbols	21
monthSymbols	21
PMSymbol	21
quarterSymbols	22
setAMSymbol:	22
setCalendar:	23
setDateFormat:	23
setDateStyle:	23

setDefaultDate:	24
setEraSymbols:	24
setFormatterBehavior:	25
setGeneratesCalendarDates:	25
setGregorianStartDate:	25
setLenient:	26
setLocale:	26
setLongEraSymbols:	27
setMonthSymbols:	27
setPMSymbol:	27
setQuarterSymbols:	28
setShortMonthSymbols:	28
setShortQuarterSymbols:	29
setShortStandaloneMonthSymbols:	29
setShortStandaloneQuarterSymbols:	30
setShortStandaloneWeekdaySymbols:	30
setShortWeekdaySymbols:	31
setStandaloneMonthSymbols:	31
setStandaloneQuarterSymbols:	31
setStandaloneWeekdaySymbols:	32
setTimeStyle:	32
setTimeZone:	33
setTwoDigitStartDate:	33
setVeryShortMonthSymbols:	34
setVeryShortStandaloneMonthSymbols:	34
setVeryShortStandaloneWeekdaySymbols:	34
setVeryShortWeekdaySymbols:	35
setWeekdaySymbols:	35
shortMonthSymbols	36
shortQuarterSymbols	36
shortStandaloneMonthSymbols	37
shortStandaloneQuarterSymbols	37
shortStandaloneWeekdaySymbols	38
shortWeekdaySymbols	38
standaloneMonthSymbols	39
standaloneQuarterSymbols	39
standaloneWeekdaySymbols	39
stringFromDate:	40
timeStyle	40
timeZone	41
twoDigitStartDate	41
veryShortMonthSymbols	42
veryShortStandaloneMonthSymbols	42
veryShortStandaloneWeekdaySymbols	42
veryShortWeekdaySymbols	43
weekdaySymbols	43

CONTENTS

Constants	44
NSDateFormatterStyle	44
NSDateFormatterBehavior	45

Document Revision History 47

Index 49

NSDateFormatter Class Reference

Inherits from	NSFormatter : NSObject
Conforms to	NSCoding (NSFormatter) NSCopying (NSFormatter) NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Data Formatting Programming Guide for Cocoa
Declared in	NSDateFormatter.h
Related sample code	Core Data HTML Store DatePicker iSpend Mountains Reminders

Overview

Instances of `NSDateFormatter` create string representations of `NSDate` (and `NSDateCalendarDate`) objects, and convert textual representations of dates and times into `NSDate` objects. You can express the representation of dates and times flexibly: “Thu 22 Dec 1994” is just as acceptable as “12/22/94.”

With Mac OS X v10.4 and later, `NSDateFormatter` has two modes of operation (or behaviors). By default, instances of `NSDateFormatter` have the same behavior as they did on Mac OS X versions 10.0 to 10.3. You can, however, configure instances (or set a default for all instances) to adopt a new behavior implemented for Mac OS X version 10.4. See *Data Formatting Programming Guide for Cocoa* for a full description of the old and new behaviors.

iPhone OS Note: iPhone OS supports only the modern 10.4+ behavior. 10.0-style methods and format strings are not available on iPhone OS.

If you initialize a formatter using `initWithDateFormat:allowNaturalLanguage:` (page 19), you are (for backwards compatibility reasons) creating an “old-style” date formatter. To use the new behavior, you initialize the formatter with `init` (page 18). If you have not set the default class behavior (see `setDefaultFormatterBehavior:` (page 13)), you send the instance a `setFormatterBehavior:` (page 25) message with the argument `NSDateFormatterBehavior10_4`. You can then set the date format as appropriate, typically using a format style as illustrated in the following code fragment.

```
// assume default behavior set for class using
// [NSDateFormatter setDefaultFormatterBehavior:NSDateFormatterBehavior10_4];

NSDateFormatter *dateFormatter = [[[NSDateFormatter alloc] init] autorelease];
[dateFormatter setDateStyle:NSDateFormatterMediumStyle];
[dateFormatter setTimeStyle:NSDateFormatterNoStyle];

NSDate *date = [NSDate dateWithTimeIntervalSinceReferenceDate:118800];
NSString *formattedDateString = [dateFormatter stringFromDate:date];
NSLog(@"formattedDateString for locale %@: %@",
      [[dateFormatter locale] localeIdentifier], formattedDateString);

// Output: formattedDateString for locale en_US: Jan 2, 2001
```

Note that the format for a given style is dependent on a user's preferences, including the locale setting.

Note also that by default the new-style formatter returns `NSDate` objects instead of `NSDateCalendarDate` objects. You can change this behavior using [setGeneratesCalendarDates:](#) (page 25).

Tasks

Initializing a Date Formatter

- [init](#) (page 18)
Initializes and returns an `NSDateFormatter` instance.
- [initWithDateFormat:allowNaturalLanguage:](#) (page 19)
Initializes and returns an `NSDateFormatter` instance that uses the Mac OS X v10.0 formatting behavior and the given date format string in its conversions.

Managing Behavior

- [allowsNaturalLanguage](#) (page 13)
Returns a Boolean value that indicates whether the receiver attempts to process dates entered as a vernacular string.
- [formatterBehavior](#) (page 16)
Returns the formatter behavior for the receiver.
- [setFormatterBehavior:](#) (page 25)
Sets the formatter behavior for the receiver.
- + [defaultFormatterBehavior](#) (page 12)
Returns the default formatting behavior for instances of the class.
- + [setDefaultFormatterBehavior:](#) (page 13)
Sets the default formatting behavior for instances of the class.
- [generatesCalendarDates](#) (page 17)
Returns a Boolean value that indicates whether the receiver generates calendar dates.
- [setGeneratesCalendarDates:](#) (page 25)
Sets whether the receiver generates calendar dates.

- [isLenient](#) (page 20)
Returns a Boolean value that indicates whether the receiver uses heuristics when parsing a string.
- [setLenient:](#) (page 26)
Sets whether the receiver uses heuristics when parsing a string.

Converting Objects

- [dateFromString:](#) (page 15)
Returns a date representation of a given string interpreted using the receiver's current settings.
- [stringFromDate:](#) (page 40)
Returns a string representation of a given date formatted using the receiver's current settings.
- [getObjectValue:forString:range:error:](#) (page 17)
Returns by reference a date representation of a given string and the range of the string used, and returns a Boolean value that indicates whether the string could be parsed.

Managing Formats and Styles

- [dateFormat](#) (page 14)
Returns the date format string used by the receiver.
- [setDateFormat:](#) (page 23)
Sets the date format for the receiver.
- [dateStyle](#) (page 15)
Returns the date style of the receiver.
- [setDateStyle:](#) (page 23)
Sets the date style of the receiver.
- [timeStyle](#) (page 40)
Returns the time style of the receiver.
- [setTimeStyle:](#) (page 32)
Sets the time style of the receiver.

Managing Attributes

- [calendar](#) (page 14)
Returns the calendar for the receiver.
- [setCalendar:](#) (page 23)
Sets the calendar for the receiver.
- [defaultDate](#) (page 16)
Returns the default date for the receiver.
- [setDefaultDate:](#) (page 24)
Sets the default date for the receiver.
- [locale](#) (page 20)
Returns the locale for the receiver.

- [setLocale:](#) (page 26)
Sets the locale for the receiver.
- [timeZone](#) (page 41)
Returns the time zone for the receiver.
- [setTimeZone:](#) (page 33)
Sets the time zone for the receiver.
- [twoDigitStartDate](#) (page 41)
Returns the earliest date that can be denoted by a two-digit year specifier.
- [setTwoDigitStartDate:](#) (page 33)
Sets the two-digit start date for the receiver.
- [gregorianStartDate](#) (page 18)
Returns the start date of the Gregorian calendar for the receiver.
- [setGregorianStartDate:](#) (page 25)
Sets the start date of the Gregorian calendar for the receiver.

Managing AM and PM Symbols

- [AMSymbol](#) (page 14)
Returns the AM symbol for the receiver.
- [setAMSymbol:](#) (page 22)
Sets the AM symbol for the receiver.
- [PMSymbol](#) (page 21)
Returns the PM symbol for the receiver.
- [setPMSymbol:](#) (page 27)
Sets the PM symbol for the receiver.

Managing Weekday Symbols

- [weekdaySymbols](#) (page 43)
Returns the array of weekday symbols for the receiver.
- [setWeekdaySymbols:](#) (page 35)
Sets the weekday symbols for the receiver.
- [shortWeekdaySymbols](#) (page 38)
Returns the array of short weekday symbols for the receiver.
- [setShortWeekdaySymbols:](#) (page 31)
Sets the short weekday symbols for the receiver.
- [veryShortWeekdaySymbols](#) (page 43)
Returns the array of very short weekday symbols for the receiver.
- [setVeryShortWeekdaySymbols:](#) (page 35)
Sets the vert short weekday symbols for the receiver
- [standaloneWeekdaySymbols](#) (page 39)
Returns the array of standalone weekday symbols for the receiver.

- [setStandaloneWeekdaySymbols:](#) (page 32)
Sets the standalone weekday symbols for the receiver.
- [shortStandaloneWeekdaySymbols](#) (page 38)
Returns the array of short standalone weekday symbols for the receiver.
- [setShortStandaloneWeekdaySymbols:](#) (page 30)
Sets the short standalone weekday symbols for the receiver.
- [veryShortStandaloneWeekdaySymbols](#) (page 42)
Returns the array of very short standalone weekday symbols for the receiver.
- [setVeryShortStandaloneWeekdaySymbols:](#) (page 34)
Sets the very short standalone weekday symbols for the receiver.

Managing Month Symbols

- [monthSymbols](#) (page 21)
Returns the month symbols for the receiver.
- [setMonthSymbols:](#) (page 27)
Sets the month symbols for the receiver.
- [shortMonthSymbols](#) (page 36)
Returns the array of short month symbols for the receiver.
- [setShortMonthSymbols:](#) (page 28)
Sets the short month symbols for the receiver.
- [veryShortMonthSymbols](#) (page 42)
Returns the very short month symbols for the receiver.
- [setVeryShortMonthSymbols:](#) (page 34)
Sets the very short month symbols for the receiver.
- [standaloneMonthSymbols](#) (page 39)
Returns the standalone month symbols for the receiver.
- [setStandaloneMonthSymbols:](#) (page 31)
Sets the standalone month symbols for the receiver.
- [shortStandaloneMonthSymbols](#) (page 37)
Returns the short standalone month symbols for the receiver.
- [setShortStandaloneMonthSymbols:](#) (page 29)
Sets the short standalone month symbols for the receiver.
- [veryShortStandaloneMonthSymbols](#) (page 42)
Returns the very short month symbols for the receiver.
- [setVeryShortStandaloneMonthSymbols:](#) (page 34)
Sets the very short standalone month symbols for the receiver.

Managing Quarter Symbols

- [quarterSymbols](#) (page 22)
Returns the quarter symbols for the receiver.

- [setQuarterSymbols:](#) (page 28)
Sets the quarter symbols for the receiver.
- [shortQuarterSymbols](#) (page 36)
Returns the short quarter symbols for the receiver.
- [setShortQuarterSymbols:](#) (page 29)
Sets the short quarter symbols for the receiver.
- [standaloneQuarterSymbols](#) (page 39)
Returns the standalone quarter symbols for the receiver.
- [setStandaloneQuarterSymbols:](#) (page 31)
Sets the standalone quarter symbols for the receiver.
- [shortStandaloneQuarterSymbols](#) (page 37)
Returns the short standalone quarter symbols for the receiver.
- [setShortStandaloneQuarterSymbols:](#) (page 30)
Sets the short standalone quarter symbols for the receiver.

Managing Era Symbols

- [eraSymbols](#) (page 16)
Returns the era symbols for the receiver.
- [setEraSymbols:](#) (page 24)
Sets the era symbols for the receiver.
- [longEraSymbols](#) (page 21)
Returns the long era symbols for the receiver
- [setLongEraSymbols:](#) (page 27)
Sets the long era symbols for the receiver.

Class Methods

defaultFormatterBehavior

Returns the default formatting behavior for instances of the class.

```
+ (NSDateFormatterBehavior)defaultFormatterBehavior
```

Return Value

The default formatting behavior for instances of the class. For possible values, see [NSDateFormatterBehavior](#) (page 45).

Discussion

The default is `NSDateFormatterBehavior10_0`.

Availability

Available in Mac OS X v10.4 and later.

See Also

+ [setDefaultFormatterBehavior:](#) (page 13).

- [formatterBehavior](#) (page 16)
- [setFormatterBehavior:](#) (page 25)

Declared In

NSDateFormatter.h

setDefaultFormatterBehavior:

Sets the default formatting behavior for instances of the class.

+ (void)setDefaultFormatterBehavior:(NSDateFormatterBehavior)*behavior***Parameters***behavior*

The default formatting behavior for instances of the class. For possible values, see [NSDateFormatterBehavior](#) (page 45).

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [defaultFormatterBehavior](#) (page 12)
- [formatterBehavior](#) (page 16)
- [setFormatterBehavior:](#) (page 25)

Related Sample Code

DatePicker

Declared In

NSDateFormatter.h

Instance Methods

allowsNaturalLanguage

Returns a Boolean value that indicates whether the receiver attempts to process dates entered as a vernacular string.

- (BOOL)allowsNaturalLanguage

Return Value

YES if the receiver attempts to process dates entered as a vernacular string ("today," "next week," "dinner time," and so on), otherwise NO.

Discussion

Natural-language processing supports only a limited set of colloquial phrases, primarily in English. It may give unexpected results, and its use is strongly discouraged.

Special ConsiderationsThis method is for use with formatters using `NSDateFormatterBehavior10_0` behavior.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSDateFormatter.h

AMSymbol

Returns the AM symbol for the receiver.

- (NSString *)AMSymbol

Return Value

The AM symbol for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAMSymbol:](#) (page 22)
- [PMSymbol](#) (page 21)
- [setPMSymbol:](#) (page 27)

Declared In

NSDateFormatter.h

calendar

Returns the calendar for the receiver.

- (NSCalendar *)calendar

Return Value

The calendar for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setCalendar:](#) (page 23)

Declared In

NSDateFormatter.h

dateFormat

Returns the date format string used by the receiver.

- (NSString *)dateFormat

Return Value

The date format string used by the receiver.

Discussion

See [Date Format String Syntax \(Mac OS X Versions 10.0 to 10.3\)](#) for a list of the conversion specifiers permitted in date format strings.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setDateFormat:](#) (page 23)

Declared In

NSDateFormatter.h

dateFromString:

Returns a date representation of a given string interpreted using the receiver's current settings.

```
- (NSDate *)dateFromString:(NSString *)string
```

Parameters

string

The string to parse.

Return Value

A date representation of *string* interpreted using the receiver's current settings.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [getObjectValue:forString:range:error:](#) (page 17)

- [stringFromDate:](#) (page 40)

Related Sample Code

Reminders

Declared In

NSDateFormatter.h

dateStyle

Returns the date style of the receiver.

```
- (NSDateFormatterStyle)dateStyle
```

Return Value

The date style of the receiver. For possible values, see [NSDateFormatterStyle](#) (page 44).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDateStyle:](#) (page 23)

Declared In

NSDateFormatter.h

defaultDate

Returns the default date for the receiver.

- (NSDate *)defaultDate

Return Value

The default date for the receiver.

Discussion

The default default date is nil.

Availability

Available in Mac OS X v10.4 and later.

See Also- [setDefaultDate:](#) (page 24)**Declared In**

NSDateFormatter.h

eraSymbols

Returns the era symbols for the receiver.

- (NSArray *)eraSymbols

Return Value

An array containing NSString objects representing the era symbols for the receiver (for example, {"B.C.E.", "C.E."}).

Availability

Available in Mac OS X v10.4 and later.

See Also- [setEraSymbols:](#) (page 24)- [longEraSymbols](#) (page 21)**Declared In**

NSDateFormatter.h

formatterBehavior

Returns the formatter behavior for the receiver.

- (NSDateFormatterBehavior)formatterBehavior

Return ValueThe formatter behavior for the receiver. For possible values, see [NSDateFormatterBehavior](#) (page 45).

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [defaultFormatterBehavior](#) (page 12).
- + [setDefaultFormatterBehavior:](#) (page 13)
- [setFormatterBehavior:](#) (page 25)

Declared In

NSDateFormatter.h

generatesCalendarDates

Returns a Boolean value that indicates whether the receiver generates calendar dates.

- (BOOL)generatesCalendarDates

Return Value

YES if the receiver generates calendar dates, otherwise NO.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setGeneratesCalendarDates:](#) (page 25)

Declared In

NSDateFormatter.h

getObjectValue:forString:range:error:

Returns by reference a date representation of a given string and the range of the string used, and returns a Boolean value that indicates whether the string could be parsed.

- (BOOL)getObjectValue:(id *)obj forString:(NSString *)string range:(inout NSRange *)rangep error:(NSError **)error

Parameters

obj

If the receiver is able to parse *string*, upon return contains a date representation of *string*.

string

The string to parse.

rangep

If the receiver is able to parse *string*, upon return contains the range of *string* used to create the date.

error

If the receiver is unable to create a date by parsing *string*, upon return contains an NSError object that describes the problem.

Return Value

YES if the receiver can create a date by parsing *string*, otherwise NO.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dateFromString:](#) (page 15)
- [stringForObjectValue:](#)

Related Sample Code

iSpend

Declared In

NSDateFormatter.h

gregorianStartDate

Returns the start date of the Gregorian calendar for the receiver.

- (NSDate *)gregorianStartDate

Return Value

The start date of the Gregorian calendar for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setGregorianStartDate:](#) (page 25)

Declared In

NSDateFormatter.h

init

Initializes and returns an NSDateFormatter instance.

- (id)init

Return Value

An NSDateFormatter instance initialized with locale, time zone, calendar, and behavior set to the appropriate default values.

Discussion

There are many new attributes you can get and set on a 10.4-style date formatter, including the locale, time zone, calendar, format string, the two-digit-year cross-over date, the default date which provides unspecified components, and there is also access to the various textual strings, like the month names. You are encouraged, however, not to change individual settings. Instead you should accept the default settings established on initialization and specify the format using [setDateStyle:](#) (page 23), [setTimeStyle:](#) (page 32), and appropriate style constants (see [NSDateFormatterStyle](#) (page 44)—these are styles that the user can configure in the International preferences panel in System Preferences).

Special Considerations

If you want the Mac OS X 10.4 behavior but have not set the class's default behavior to `NSDateFormatterBehavior10_4`, you also need to send the new instance a `setFormatterBehavior:` (page 25) message with the argument `NSDateFormatterBehavior10_4`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `initWithDateFormat:allowNaturalLanguage:` (page 19)
- `setDateStyle:` (page 23)
- `setTimeStyle:` (page 32)

Declared In

`NSDateFormatter.h`

`initWithDateFormat:allowNaturalLanguage:`

Initializes and returns an `NSDateFormatter` instance that uses the Mac OS X v10.0 formatting behavior and the given date format string in its conversions.

```
- (id)initWithDateFormat:(NSString *)format allowNaturalLanguage:(BOOL)flag
```

Parameters

format

The format for the receiver. See Date Format String Syntax (Mac OS X Versions 10.0 to 10.3) for a list of conversion specifiers permitted in date format strings.

flag

A flag that specifies whether the receiver should process dates entered as expressions in the vernacular (for example, "tomorrow")—YES means that it should.

Return Value

An initialized `NSDateFormatter` instance that uses *format* in its conversions and that uses the Mac OS X v10.0 formatting behavior.

Discussion

`NSDateFormatter` attempts natural-language processing only after it fails to interpret an entered string according to *format*. Natural-language processing supports only a limited set of colloquial phrases, primarily in English. It may give unexpected results, and its use is strongly discouraged.

The following example creates a date formatter with the format string (for example) "Mar 15 1994" and then associates the formatter with the cells of a form (`contactsForm`):

```
NSDateFormatter *dateFormat = [[NSDateFormatter alloc]
    initWithDateFormat:@"%b %d %Y" allowNaturalLanguage:NO];
[[contactsForm cells] makeObjectsPerformSelector:@selector(setFormatter:)
    withObject:dateFormat];
```

Important: You cannot use this method to initialize a formatter with the Mac OS X v10.4 formatting behavior, you must use `initWithFormat` (page 18).

Availability

Available in Mac OS X v10.0 and later.

See Also

- `initWithFormat` (page 18)

Declared In

NSDateFormatter.h

isLenient

Returns a Boolean value that indicates whether the receiver uses heuristics when parsing a string.

- (BOOL)isLenient

Return Value

YES if the receiver has been set to use heuristics when parsing a string to guess at the date which is intended, otherwise NO.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `setLenient:` (page 26)

Declared In

NSDateFormatter.h

locale

Returns the locale for the receiver.

- (NSLocale *)locale

Return Value

The locale for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `setLocale:` (page 26)

Declared In

NSDateFormatter.h

longEraSymbols

Returns the long era symbols for the receiver.

- (NSArray *)longEraSymbols

Return Value

An array containing `NSString` objects representing the era symbols for the receiver (for example, {"Before Common Era", "Common Era"}).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setLongEraSymbols:](#) (page 27)
- [eraSymbols](#) (page 16)

Declared In

NSDateFormatter.h

monthSymbols

Returns the month symbols for the receiver.

- (NSArray *)monthSymbols

Return Value

An array of `NSString` objects that specify the month symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setMonthSymbols:](#) (page 27)
- [shortMonthSymbols](#) (page 36)
- [veryShortMonthSymbols](#) (page 42)
- [standaloneMonthSymbols](#) (page 39)
- [shortStandaloneMonthSymbols](#) (page 37)
- [veryShortStandaloneMonthSymbols](#) (page 42)

Declared In

NSDateFormatter.h

PMSymbol

Returns the PM symbol for the receiver.

- (NSString *)PMSymbol

Return Value

The PM symbol for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setPMSymbol:](#) (page 27)
- [AMSymbol](#) (page 14)
- [setAMSymbol:](#) (page 22)

Declared In

NSDateFormatter.h

quarterSymbols

Returns the quarter symbols for the receiver.

- (NSArray *)quarterSymbols

Return Value

An array containing NSString objects representing the quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setQuarterSymbols:](#) (page 28)
- [shortQuarterSymbols](#) (page 36)
- [standaloneQuarterSymbols](#) (page 39)
- [shortStandaloneQuarterSymbols](#) (page 37)

Declared In

NSDateFormatter.h

setAMSymbol:

Sets the AM symbol for the receiver.

- (void)setAMSymbol:(NSString *)*string*

Parameters

string

The AM symbol for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [AMSymbol](#) (page 14)
- [PMSymbol](#) (page 21)
- [setPMSymbol:](#) (page 27)

Declared In

NSDateFormatter.h

setCalendar:

Sets the calendar for the receiver.

```
- (void)setCalendar:(NSCalendar *)calendar
```

Parameters

calendar

The calendar for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [calendar](#) (page 14)

Declared In

NSDateFormatter.h

setDateFormat:

Sets the date format for the receiver.

```
- (void)setDateFormat:(NSString *)string
```

Parameters

string

The date format for the receiver. See *Data Formatting Programming Guide for Cocoa* for a list of the conversion specifiers permitted in date format strings.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dateFormat](#) (page 14).

Declared In

NSDateFormatter.h

setDateStyle:

Sets the date style of the receiver.

```
- (void)setDateStyle:(NSDateFormatterStyle)style
```

Parameters

style

The date style of the receiver. For possible values, see [NSDateFormatterStyle](#) (page 44).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dateStyle](#) (page 15).

Related Sample Code

DatePicker

iSpend

Mountains

NSOperationSample

Reminders

Declared In

NSDateFormatter.h

setDefaultDate:

Sets the default date for the receiver.

- (void)setDefaultDate:(NSDate *)*date***Parameters***date*

The default date for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also- [defaultDate](#) (page 16)**Declared In**

NSDateFormatter.h

setEraSymbols:

Sets the era symbols for the receiver.

- (void)setEraSymbols:(NSArray *)*array***Parameters***array*

An array containing NSString objects representing the era symbols for the receiver (for example, {"B.C.E.,"C.E."}).

Availability

Available in Mac OS X v10.4 and later.

See Also- [eraSymbols](#) (page 16)- [longEraSymbols](#) (page 21)**Declared In**

NSDateFormatter.h

setFormatterBehavior:

Sets the formatter behavior for the receiver.

```
- (void)setFormatterBehavior:(NSDateFormatterBehavior)behavior
```

Parameters

behavior

The formatter behavior for the receiver. For possible values, see [NSDateFormatterBehavior](#) (page 45).

Availability

Available in Mac OS X v10.4 and later.

See Also

- + [defaultFormatterBehavior](#) (page 12).
- + [setDefaultFormatterBehavior:](#) (page 13)
- [formatterBehavior](#) (page 16)

Declared In

NSDateFormatter.h

setGeneratesCalendarDates:

Sets whether the receiver generates calendar dates.

```
- (void)setGeneratesCalendarDates:(BOOL)b
```

Parameters

b

A Boolean value that specifies whether the receiver generates calendar dates.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [generatesCalendarDates](#) (page 17).

Declared In

NSDateFormatter.h

setGregorianStartDate:

Sets the start date of the Gregorian calendar for the receiver.

```
- (void)setGregorianStartDate:(NSDate *)array
```

Parameters

array

The start date of the Gregorian calendar for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [gregorianStartDate](#) (page 18)

Declared In

NSDateFormatter.h

setLenient:

Sets whether the receiver uses heuristics when parsing a string.

```
- (void)setLenient:(BOOL)b
```

Parameters

b

YES to use heuristics when parsing a string to guess at the date which is intended, otherwise NO.

Discussion

If a formatter is set to be lenient, when parsing a string it uses heuristics to guess at the date which is intended. As with any guessing, it may get the result date wrong (that is, a date other than that which was intended).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isLenient](#) (page 20)

Declared In

NSDateFormatter.h

setLocale:

Sets the locale for the receiver.

```
- (void)setLocale:(NSLocale *)locale
```

Parameters

locale

The locale for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [locale](#) (page 20)

Related Sample Code

Mountains

Declared In

NSDateFormatter.h

setLongEraSymbols:

Sets the long era symbols for the receiver.

```
- (void)setLongEraSymbols:(NSArray *)array
```

Parameters

array

An array containing NSString objects representing the era symbols for the receiver (for example, {"Before Common Era", "Common Era"}).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [longEraSymbols](#) (page 21)
- [eraSymbols](#) (page 16)

Declared In

NSDateFormatter.h

setMonthSymbols:

Sets the month symbols for the receiver.

```
- (void)setMonthSymbols:(NSArray *)array
```

Parameters

array

An array of NSString objects that specify the month symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [monthSymbols](#) (page 21)
- [setShortMonthSymbols:](#) (page 28)
- [setVeryShortMonthSymbols:](#) (page 34)
- [setStandaloneMonthSymbols:](#) (page 31)
- [setShortStandaloneMonthSymbols:](#) (page 29)
- [setVeryShortStandaloneMonthSymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setPMSymbol:

Sets the PM symbol for the receiver.

```
- (void)setPMSymbol:(NSString *)string
```

Parameters

string

The PM symbol for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [PMSymbol](#) (page 21)
- [AMSymbol](#) (page 14)
- [setAMSymbol:](#) (page 22)

Declared In

NSDateFormatter.h

setQuarterSymbols:

Sets the quarter symbols for the receiver.

```
- (void)setQuarterSymbols:(NSArray *)array
```

Parameters

array

An array of NSString objects that specify the quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [quarterSymbols](#) (page 22)
- [setShortQuarterSymbols:](#) (page 29)
- [setStandaloneQuarterSymbols:](#) (page 31)
- [setShortStandaloneQuarterSymbols:](#) (page 30)

Declared In

NSDateFormatter.h

setShortMonthSymbols:

Sets the short month symbols for the receiver.

```
- (void)setShortMonthSymbols:(NSArray *)array
```

Parameters

array

An array of NSString objects that specify the short month symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [shortMonthSymbols](#) (page 36)

- [setMonthSymbols:](#) (page 27)
- [setVeryShortMonthSymbols:](#) (page 34)
- [setStandaloneMonthSymbols:](#) (page 31)
- [setShortStandaloneMonthSymbols:](#) (page 29)
- [setVeryShortStandaloneMonthSymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setShortQuarterSymbols:

Sets the short quarter symbols for the receiver.

- (void)setShortQuarterSymbols:(NSArray *)array

Parameters

array

An array of NSString objects that specify the short quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [shortQuarterSymbols](#) (page 36)
- [setQuarterSymbols:](#) (page 28)
- [setStandaloneQuarterSymbols:](#) (page 31)
- [setShortStandaloneQuarterSymbols:](#) (page 30)

Declared In

NSDateFormatter.h

setShortStandaloneMonthSymbols:

Sets the short standalone month symbols for the receiver.

- (void)setShortStandaloneMonthSymbols:(NSArray *)array

Parameters

array

An array of NSString objects that specify the short standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [shortStandaloneMonthSymbols](#) (page 37)
- [setMonthSymbols:](#) (page 27)
- [setShortMonthSymbols:](#) (page 28)
- [setVeryShortMonthSymbols:](#) (page 34)
- [setStandaloneMonthSymbols:](#) (page 31)
- [setVeryShortStandaloneMonthSymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setShortStandaloneQuarterSymbols:

Sets the short standalone quarter symbols for the receiver.

- (void)setShortStandaloneQuarterSymbols:(NSArray *)array

Parameters*array*

An array of NSString objects that specify the short standalone quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [shortStandaloneQuarterSymbols](#) (page 37)
- [setQuarterSymbols:](#) (page 28)
- [setShortQuarterSymbols:](#) (page 29)
- [setStandaloneQuarterSymbols:](#) (page 31)

Declared In

NSDateFormatter.h

setShortStandaloneWeekdaySymbols:

Sets the short standalone weekday symbols for the receiver.

- (void)setShortStandaloneWeekdaySymbols:(NSArray *)array

Parameters*array*

An array of NSString objects that specify the short standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [shortStandaloneWeekdaySymbols](#) (page 38)
- [setWeekdaySymbols:](#) (page 35)
- [setShortWeekdaySymbols:](#) (page 31)
- [setVeryShortWeekdaySymbols:](#) (page 35)
- [setStandaloneWeekdaySymbols:](#) (page 32)
- [setVeryShortStandaloneWeekdaySymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setShortWeekdaySymbols:

Sets the short weekday symbols for the receiver.

```
- (void)setShortWeekdaySymbols:(NSArray *)array
```

Parameters

array

An array of `NSString` objects that specify the short weekday symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [shortWeekdaySymbols](#) (page 38)
- [setWeekdaySymbols](#): (page 35)
- [setVeryShortWeekdaySymbols](#): (page 35)
- [setStandaloneWeekdaySymbols](#): (page 32)
- [setShortStandaloneWeekdaySymbols](#): (page 30)
- [setVeryShortStandaloneWeekdaySymbols](#): (page 34)

Declared In

NSDateFormatter.h

setStandaloneMonthSymbols:

Sets the standalone month symbols for the receiver.

```
- (void)setStandaloneMonthSymbols:(NSArray *)array
```

Parameters

array

An array of `NSString` objects that specify the standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [standaloneMonthSymbols](#) (page 39)
- [setMonthSymbols](#): (page 27)
- [setShortMonthSymbols](#): (page 28)
- [setVeryShortMonthSymbols](#): (page 34)
- [setShortStandaloneMonthSymbols](#): (page 29)
- [setVeryShortStandaloneMonthSymbols](#): (page 34)

Declared In

NSDateFormatter.h

setStandaloneQuarterSymbols:

Sets the standalone quarter symbols for the receiver.

- (void)setStandaloneQuarterSymbols:(NSArray *)array

Parameters

array

An array of NSString objects that specify the standalone quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setStandaloneQuarterSymbols:](#) (page 31)
- [setQuarterSymbols:](#) (page 28)
- [setShortQuarterSymbols:](#) (page 29)
- [setShortStandaloneQuarterSymbols:](#) (page 30)

Declared In

NSDateFormatter.h

setStandaloneWeekdaySymbols:

Sets the standalone weekday symbols for the receiver.

- (void)setStandaloneWeekdaySymbols:(NSArray *)array

Parameters

array

An array of NSString objects that specify the standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [standaloneWeekdaySymbols](#) (page 39)
- [setWeekdaySymbols:](#) (page 35)
- [setShortWeekdaySymbols:](#) (page 31)
- [setVeryShortWeekdaySymbols:](#) (page 35)
- [setShortStandaloneWeekdaySymbols:](#) (page 30)
- [setVeryShortStandaloneWeekdaySymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setTimeStyle:

Sets the time style of the receiver.

- (void)setTimeStyle:(NSDateFormatterStyle)style

Parameters

style

The time style for the receiver. For possible values, see [NSDateFormatterStyle](#) (page 44).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [timeStyle](#) (page 40)

Related Sample Code

[DatePicker](#)

[Mountains](#)

[NSOperationSample](#)

[Reminders](#)

Declared In

`NSDateFormatter.h`

setTimeZone:

Sets the time zone for the receiver.

```
- (void)setTimeZone:(NSTimeZone *)tz
```

Parameters

tz

The time zone for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [timeZone](#) (page 41)

Declared In

`NSDateFormatter.h`

setTwoDigitStartDate:

Sets the two-digit start date for the receiver.

```
- (void)setTwoDigitStartDate:(NSDate *)date
```

Parameters

date

The earliest date that can be denoted by a two-digit year specifier.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [twoDigitStartDate](#) (page 41)

Declared In

`NSDateFormatter.h`

setVeryShortMonthSymbols:

Sets the very short month symbols for the receiver.

```
- (void)setVeryShortMonthSymbols:(NSArray *)array
```

Parameters

array

An array of `NSString` objects that specify the very short month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [veryShortMonthSymbols](#) (page 42)
- [setMonthSymbols:](#) (page 27)
- [setShortMonthSymbols:](#) (page 28)
- [setStandaloneMonthSymbols:](#) (page 31)
- [setShortStandaloneMonthSymbols:](#) (page 29)
- [setVeryShortStandaloneMonthSymbols:](#) (page 34)

Declared In

NSDateFormatter.h

setVeryShortStandaloneMonthSymbols:

Sets the very short standalone month symbols for the receiver.

```
- (void)setVeryShortStandaloneMonthSymbols:(NSArray *)array
```

Parameters

array

An array of `NSString` objects that specify the very short standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [veryShortStandaloneMonthSymbols](#) (page 42)
- [setMonthSymbols:](#) (page 27)
- [setShortMonthSymbols:](#) (page 28)
- [setVeryShortMonthSymbols:](#) (page 34)
- [setStandaloneMonthSymbols:](#) (page 31)
- [setShortStandaloneMonthSymbols:](#) (page 29)

Declared In

NSDateFormatter.h

setVeryShortStandaloneWeekdaySymbols:

Sets the very short standalone weekday symbols for the receiver.

- (void)setVeryShortStandaloneWeekdaySymbols:(NSArray *)array

Parameters

array

An array of `NSString` objects that specify the very short standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [veryShortStandaloneWeekdaySymbols](#) (page 42)
- [setWeekdaySymbols](#): (page 35)
- [setShortWeekdaySymbols](#): (page 31)
- [setVeryShortWeekdaySymbols](#): (page 35)
- [setStandaloneWeekdaySymbols](#): (page 32)
- [setShortStandaloneWeekdaySymbols](#): (page 30)

Declared In

NSDateFormatter.h

setVeryShortWeekdaySymbols:

Sets the vert short weekday symbols for the receiver

- (void)setVeryShortWeekdaySymbols:(NSArray *)array

Parameters

array

An array of `NSString` objects that specify the very short weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [veryShortWeekdaySymbols](#) (page 43)
- [setWeekdaySymbols](#): (page 35)
- [setShortWeekdaySymbols](#): (page 31)
- [setStandaloneWeekdaySymbols](#): (page 32)
- [setShortStandaloneWeekdaySymbols](#): (page 30)
- [setVeryShortStandaloneWeekdaySymbols](#): (page 34)

Declared In

NSDateFormatter.h

setWeekdaySymbols:

Sets the weekday symbols for the receiver.

- (void)setWeekdaySymbols:(NSArray *)array

Parameters*array*

An array of `NSString` objects that specify the weekday symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [weekdaySymbols](#) (page 43)
- [setShortWeekdaySymbols:](#) (page 31)
- [setVeryShortWeekdaySymbols:](#) (page 35)
- [setStandaloneWeekdaySymbols:](#) (page 32)
- [setShortStandaloneWeekdaySymbols:](#) (page 30)
- [setVeryShortStandaloneWeekdaySymbols:](#) (page 34)

Declared In

NSDateFormatter.h

shortMonthSymbols

Returns the array of short month symbols for the receiver.

- (NSArray *)shortMonthSymbols

Return Value

An array containing `NSString` objects representing the short month symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setShortMonthSymbols:](#) (page 28)
- [monthSymbols](#) (page 21)
- [veryShortMonthSymbols](#) (page 42)
- [standaloneMonthSymbols](#) (page 39)
- [shortStandaloneMonthSymbols](#) (page 37)
- [veryShortStandaloneMonthSymbols](#) (page 42)

Declared In

NSDateFormatter.h

shortQuarterSymbols

Returns the short quarter symbols for the receiver.

- (NSArray *)shortQuarterSymbols

Return Value

An array containing `NSString` objects representing the short quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShortQuarterSymbols:](#) (page 29)
- [quarterSymbols](#) (page 22)
- [standaloneQuarterSymbols](#) (page 39)
- [shortStandaloneQuarterSymbols](#) (page 37)

Declared In

NSDateFormatter.h

shortStandaloneMonthSymbols

Returns the short standalone month symbols for the receiver.

- (NSArray *)shortStandaloneMonthSymbols

Return Value

An array of NSString objects that specify the short standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShortStandaloneMonthSymbols:](#) (page 29)
- [monthSymbols](#) (page 21)
- [shortMonthSymbols](#) (page 36)
- [veryShortMonthSymbols](#) (page 42)
- [standaloneMonthSymbols](#) (page 39)
- [veryShortStandaloneMonthSymbols](#) (page 42)

Declared In

NSDateFormatter.h

shortStandaloneQuarterSymbols

Returns the short standalone quarter symbols for the receiver.

- (NSArray *)shortStandaloneQuarterSymbols

Return Value

An array containing NSString objects representing the short standalone quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShortStandaloneQuarterSymbols:](#) (page 30)
- [quarterSymbols](#) (page 22)
- [shortQuarterSymbols](#) (page 36)

- [standaloneQuarterSymbols](#) (page 39)

Declared In

NSDateFormatter.h

shortStandaloneWeekdaySymbols

Returns the array of short standalone weekday symbols for the receiver.

- (NSArray *)shortStandaloneWeekdaySymbols

Return Value

An array of NSString objects that specify the short standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShortStandaloneWeekdaySymbols:](#) (page 30)
- [weekdaySymbols](#) (page 43)
- [shortWeekdaySymbols](#) (page 38)
- [veryShortWeekdaySymbols](#) (page 43)
- [standaloneWeekdaySymbols](#) (page 39)
- [veryShortStandaloneWeekdaySymbols](#) (page 42)

Declared In

NSDateFormatter.h

shortWeekdaySymbols

Returns the array of short weekday symbols for the receiver.

- (NSArray *)shortWeekdaySymbols

Return Value

An array of NSString objects that specify the short weekday symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setShortWeekdaySymbols:](#) (page 31)
- [weekdaySymbols](#) (page 43)
- [veryShortWeekdaySymbols](#) (page 43)
- [standaloneWeekdaySymbols](#) (page 39)
- [shortStandaloneWeekdaySymbols](#) (page 38)
- [veryShortStandaloneWeekdaySymbols](#) (page 42)

Declared In

NSDateFormatter.h

standaloneMonthSymbols

Returns the standalone month symbols for the receiver.

- (NSArray *)standaloneMonthSymbols

Return Value

An array of NSString objects that specify the standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [monthSymbols](#) (page 21)
- [setStandaloneMonthSymbols:](#) (page 31)
- [shortMonthSymbols](#) (page 36)
- [veryShortMonthSymbols](#) (page 42)
- [shortStandaloneMonthSymbols](#) (page 37)
- [veryShortStandaloneMonthSymbols](#) (page 42)

Declared In

NSDateFormatter.h

standaloneQuarterSymbols

Returns the standalone quarter symbols for the receiver.

- (NSArray *)standaloneQuarterSymbols

Return Value

An array containing NSString objects representing the standalone quarter symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setStandaloneQuarterSymbols:](#) (page 31)
- [quarterSymbols](#) (page 22)
- [shortQuarterSymbols](#) (page 36)
- [shortStandaloneQuarterSymbols](#) (page 37)

Declared In

NSDateFormatter.h

standaloneWeekdaySymbols

Returns the array of standalone weekday symbols for the receiver.

- (NSArray *)standaloneWeekdaySymbols

Return Value

An array of NSString objects that specify the standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setStandaloneWeekdaySymbols:](#) (page 32)
- [weekdaySymbols](#) (page 43)
- [shortWeekdaySymbols](#) (page 38)
- [veryShortWeekdaySymbols](#) (page 43)
- [shortStandaloneWeekdaySymbols](#) (page 38)
- [veryShortStandaloneWeekdaySymbols](#) (page 42)

Declared In

NSDateFormatter.h

stringFromDate:

Returns a string representation of a given date formatted using the receiver's current settings.

```
- (NSString *)stringFromDate:(NSDate *)date
```

Parameters

date

The date to format.

Return Value

A string representation of *date* formatted using the receiver's current settings.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dateFromString:](#) (page 15)

Related Sample Code

DatePicker

iSpend

Mountains

NSOperationSample

Reminders

Declared In

NSDateFormatter.h

timeStyle

Returns the time style of the receiver.

```
- (NSDateFormatterStyle)timeStyle
```

Return Value

The time style of the receiver. For possible values, see [NSDateFormatterStyle](#) (page 44).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setTimeStyle:](#) (page 32)

Declared In

NSDateFormatter.h

timeZone

Returns the time zone for the receiver.

- (NSTimeZone *)timeZone

Return Value

The time zone for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setTimeZone:](#) (page 33)

Declared In

NSDateFormatter.h

twoDigitStartDate

Returns the earliest date that can be denoted by a two-digit year specifier.

- (NSDate *)twoDigitStartDate

Return Value

The earliest date that can be denoted by a two-digit year specifier.

Discussion

If the two-digit start date is set to January 6, 1976, then “January 1, 76” is interpreted as New Year’s Day in 2076, whereas “February 14, 76” is interpreted as Valentine’s Day in 1976.

The default date is December 31, 1949.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setTwoDigitStartDate:](#) (page 33)

Declared In

NSDateFormatter.h

veryShortMonthSymbols

Returns the very short month symbols for the receiver.

- (NSArray *)veryShortMonthSymbols

Return Value

An array of NSString objects that specify the very short month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setVeryShortMonthSymbols:](#) (page 34)
- [monthSymbols](#) (page 21)
- [shortMonthSymbols](#) (page 36)
- [standaloneMonthSymbols](#) (page 39)
- [shortStandaloneMonthSymbols](#) (page 37)
- [veryShortStandaloneMonthSymbols](#) (page 42)

Declared In

NSDateFormatter.h

veryShortStandaloneMonthSymbols

Returns the very short month symbols for the receiver.

- (NSArray *)veryShortStandaloneMonthSymbols

Return Value

An array of NSString objects that specify the very short standalone month symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setVeryShortStandaloneMonthSymbols:](#) (page 34)
- [monthSymbols](#) (page 21)
- [shortMonthSymbols](#) (page 36)
- [veryShortMonthSymbols](#) (page 42)
- [standaloneMonthSymbols](#) (page 39)
- [shortStandaloneMonthSymbols](#) (page 37)

Declared In

NSDateFormatter.h

veryShortStandaloneWeekdaySymbols

Returns the array of very short standalone weekday symbols for the receiver.

- (NSArray *)veryShortStandaloneWeekdaySymbols

Return Value

An array of `NSString` objects that specify the very short standalone weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShortStandaloneWeekdaySymbols:](#) (page 30)
- [weekdaySymbols](#) (page 43)
- [shortWeekdaySymbols](#) (page 38)
- [veryShortWeekdaySymbols](#) (page 43)
- [standaloneWeekdaySymbols](#) (page 39)
- [shortStandaloneWeekdaySymbols](#) (page 38)

Declared In

`NSDateFormatter.h`

veryShortWeekdaySymbols

Returns the array of very short weekday symbols for the receiver.

- (NSArray *)veryShortWeekdaySymbols

Return Value

An array of `NSString` objects that specify the very short weekday symbols for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setVeryShortWeekdaySymbols:](#) (page 35)
- [weekdaySymbols](#) (page 43)
- [shortWeekdaySymbols](#) (page 38)
- [standaloneWeekdaySymbols](#) (page 39)
- [shortStandaloneWeekdaySymbols](#) (page 38)
- [veryShortStandaloneWeekdaySymbols](#) (page 42)

Declared In

`NSDateFormatter.h`

weekdaySymbols

Returns the array of weekday symbols for the receiver.

- (NSArray *)weekdaySymbols

Return Value

An array of `NSString` objects that specify the weekday symbols for the receiver.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setWeekdaySymbols](#): (page 35)
- [shortWeekdaySymbols](#) (page 38)
- [veryShortWeekdaySymbols](#) (page 43)
- [standaloneWeekdaySymbols](#) (page 39)
- [shortStandaloneWeekdaySymbols](#) (page 38)
- [veryShortStandaloneWeekdaySymbols](#) (page 42)

Declared In

NSDateFormatter.h

Constants

NSDateFormatterStyle

The following constants specify predefined date and time format styles.

```
typedef enum {
    NSDateFormatterNoStyle      = kCFDateFormatterNoStyle,
    NSDateFormatterShortStyle   = kCFDateFormatterShortStyle,
    NSDateFormatterMediumStyle  = kCFDateFormatterMediumStyle,
    NSDateFormatterLongStyle    = kCFDateFormatterLongStyle,
    NSDateFormatterFullStyle    = kCFDateFormatterFullStyle
} NSDateFormatterStyle;
```

Constants

NSDateFormatterNoStyle

Specifies no style.

Equal to kCFDateFormatterNoStyle.

Available in Mac OS X v10.4 and later.

Declared in NSDateFormatter.h.

NSDateFormatterShortStyle

Specifies a short style, typically numeric only, such as "11/23/37" or "3:30pm".

Equal to kCFDateFormatterShortStyle.

Available in Mac OS X v10.4 and later.

Declared in NSDateFormatter.h.

NSDateFormatterMediumStyle

Specifies a medium style, typically with abbreviated text, such as "Nov 23, 1937".

Equal to kCFDateFormatterMediumStyle.

Available in Mac OS X v10.4 and later.

Declared in NSDateFormatter.h.

`NSDateFormatterLongStyle`

Specifies a long style, typically with full text, such as “November 23, 1937” or “3:30:32pm”.

Equal to `kCFDateFormatterLongStyle`.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

`NSDateFormatterFullStyle`

Specifies a full style with complete details, such as “Tuesday, April 12, 1952 AD” or “3:30:42pm PST”.

Equal to `kCFDateFormatterFullStyle`.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

Discussion

The format for these date and time styles is not exact because they depend on the locale, user preference settings, and the operating system version. Do not use these constants if you want an exact format.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSDateFormatter.h`

NSDateFormatterBehavior

Constants that specify the behavior `NSDateFormatter` should exhibit.

```
typedef enum {
    NSDateFormatterBehaviorDefault = 0,
    NSDateFormatterBehavior10_0   = 1000,
    NSDateFormatterBehavior10_4   = 1040,
} NSDateFormatterBehavior;
```

Constants

`NSDateFormatterBehaviorDefault`

Specifies default formatting behavior.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

`NSDateFormatterBehavior10_0`

Specifies formatting behavior equivalent to that in Mac OS X 10.0.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

`NSDateFormatterBehavior10_4`

Specifies formatting behavior equivalent for Mac OS X 10.4.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSDateFormatter.h`

Document Revision History

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

Date	Notes
2008-11-19	Added note about supported behaviors for iPhone OS.
2007-02-08	Clarified the meaning of "lenient."
2007-01-23	Included API introduced in Mac OS X v10.5.
2006-05-23	Clarified configuration of formatter after init.
	Clarified configuration of formatter after init.
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`allowsNaturalLanguage` [instance method 13](#)
`AMSymbol` [instance method 14](#)

C

`calendar` [instance method 14](#)

D

`dateFormat` [instance method 14](#)
`dateFromString:` [instance method 15](#)
`dateStyle` [instance method 15](#)
`defaultDate` [instance method 16](#)
`defaultFormatterBehavior` [class method 12](#)

E

`eraSymbols` [instance method 16](#)

F

`formatterBehavior` [instance method 16](#)

G

`generatesCalendarDates` [instance method 17](#)
`getObjectValue:forString:range:error:` [instance method 17](#)
`gregorianStartDate` [instance method 18](#)

I

`init` [instance method 18](#)
`initWithDateFormat:allowNaturalLanguage:` [instance method 19](#)
`isLenient` [instance method 20](#)

L

`locale` [instance method 20](#)
`longEraSymbols` [instance method 21](#)

M

`monthSymbols` [instance method 21](#)

N

`NSDateFormatterBehavior` [data type 45](#)
`NSDateFormatterBehavior10_0` [constant 45](#)
`NSDateFormatterBehavior10_4` [constant 45](#)
`NSDateFormatterBehaviorDefault` [constant 45](#)
`NSDateFormatterFullStyle` [constant 45](#)
`NSDateFormatterLongStyle` [constant 45](#)
`NSDateFormatterMediumStyle` [constant 44](#)
`NSDateFormatterNoStyle` [constant 44](#)
`NSDateFormatterShortStyle` [constant 44](#)
`NSDateFormatterStyle` [data type 44](#)

P

`PSymbol` [instance method 21](#)

Q

quarterSymbols [instance method 22](#)

S

setAMSymbol: [instance method 22](#)
 setCalendar: [instance method 23](#)
 setDateFormat: [instance method 23](#)
 setDateStyle: [instance method 23](#)
 setDefaultDate: [instance method 24](#)
 setDefaultFormatterBehavior: [class method 13](#)
 setEraSymbols: [instance method 24](#)
 setFormatterBehavior: [instance method 25](#)
 setGeneratesCalendarDates: [instance method 25](#)
 setGregorianStartDate: [instance method 25](#)
 setLenient: [instance method 26](#)
 setLocale: [instance method 26](#)
 setLongEraSymbols: [instance method 27](#)
 setMonthSymbols: [instance method 27](#)
 setPMSymbol: [instance method 27](#)
 setQuarterSymbols: [instance method 28](#)
 setShortMonthSymbols: [instance method 28](#)
 setShortQuarterSymbols: [instance method 29](#)
 setShortStandaloneMonthSymbols: [instance method 29](#)
 setShortStandaloneQuarterSymbols: [instance method 30](#)
 setShortStandaloneWeekdaySymbols: [instance method 30](#)
 setShortWeekdaySymbols: [instance method 31](#)
 setStandaloneMonthSymbols: [instance method 31](#)
 setStandaloneQuarterSymbols: [instance method 31](#)
 setStandaloneWeekdaySymbols: [instance method 32](#)
 setTimeStyle: [instance method 32](#)
 setTimeZone: [instance method 33](#)
 setTwoDigitStartDate: [instance method 33](#)
 setVeryShortMonthSymbols: [instance method 34](#)
 setVeryShortStandaloneMonthSymbols: [instance method 34](#)
 setVeryShortStandaloneWeekdaySymbols: [instance method 34](#)
 setVeryShortWeekdaySymbols: [instance method 35](#)
 setWeekdaySymbols: [instance method 35](#)
 shortMonthSymbols [instance method 36](#)
 shortQuarterSymbols [instance method 36](#)
 shortStandaloneMonthSymbols [instance method 37](#)
 shortStandaloneQuarterSymbols [instance method 37](#)
 shortStandaloneWeekdaySymbols [instance method 38](#)
 shortWeekdaySymbols [instance method 38](#)

standaloneMonthSymbols [instance method 39](#)
 standaloneQuarterSymbols [instance method 39](#)
 standaloneWeekdaySymbols [instance method 39](#)
 stringFromDate: [instance method 40](#)

T

timeStyle [instance method 40](#)
 timeZone [instance method 41](#)
 twoDigitStartDate [instance method 41](#)

V

veryShortMonthSymbols [instance method 42](#)
 veryShortStandaloneMonthSymbols [instance method 42](#)
 veryShortStandaloneWeekdaySymbols [instance method 42](#)
 veryShortWeekdaySymbols [instance method 43](#)

W

weekdaySymbols [instance method 43](#)