

---

# NSDateFormatter Reference

Core Foundation



2007-05-23



Apple Inc.  
© 2003, 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, 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 Reference 5**

---

Overview	5
Functions by Task	5
Creating a Date Formatter	5
Configuring a Date Formatter	5
Parsing Strings	6
Creating Strings From Data	6
Getting Information About a Date Formatter	6
Getting the NSDateFormatter Type ID	6
Functions	6
NSDateFormatterCopyProperty	6
NSDateFormatterCreate	7
NSDateFormatterCreateDateFromString	8
NSDateFormatterCreateStringWithAbsoluteTime	9
NSDateFormatterCreateStringWithDate	9
NSDateFormatterGetAbsoluteTimeFromString	10
NSDateFormatterGetDateStyle	11
NSDateFormatterGetFormat	11
NSDateFormatterGetLocale	11
NSDateFormatterGetTimeStyle	12
NSDateFormatterGetTypeID	12
NSDateFormatterSetFormat	12
NSDateFormatterSetProperty	13
Data Types	14
NSDateFormatterRef	14
NSDateFormatterStyle	14
Constants	14
Date Formatter Styles	14
Date Formatter Property Keys	16
Calendar Names	19

---

## **Document Revision History 21**

---

## **Index 23**

---



# CFDateFormatter Reference

---

<b>Derived From:</b>	CType
<b>Framework:</b>	CoreFoundation/CoreFoundation.h
<b>Companion guide</b>	Data Formatting Guide for Core Foundation
<b>Declared in</b>	CFDateFormatter.h CFLocale.h

## Overview

CFDateFormatter objects format the textual representations of CFDate and CFAbsoluteTime objects, and convert textual representations of dates and times into CFDate and CFAbsoluteTime objects. You can express the representation of dates and times very flexibly, for example “Thu 22 Dec 1994” is just as acceptable as “12/22/94.” You specify how strings are formatted and parsed by setting a format string and other properties of a CFDateFormatter object. The format of the format string itself is defined by [Unicode Technical Standard #35](#).

The CFDateFormatter opaque type is available in Mac OS X v10.3 and later.

## Functions by Task

### Creating a Date Formatter

[CFDateFormatterCreate](#) (page 7)

Creates a new CFDateFormatter object, localized to the given locale, which will format dates to the given date and time styles.

### Configuring a Date Formatter

[CFDateFormatterSetFormat](#) (page 12)

Sets the format string of the given date formatter to the specified value.

[CFDateFormatterSetProperty](#) (page 13)

Sets a date formatter property using a key-value pair.

## Parsing Strings

[CFDateFormatterCreateDateFromString](#) (page 8)

Returns a date object representing a given string.

[CFDateFormatterGetAbsoluteTimeFromString](#) (page 10)

Returns an absolute time object representing a given string.

## Creating Strings From Data

[CFDateFormatterCreateStringWithAbsoluteTime](#) (page 9)

Returns a string representation of the given absolute time using the specified date formatter.

[CFDateFormatterCreateStringWithDate](#) (page 9)

Returns a string representation of the given date using the specified date formatter.

## Getting Information About a Date Formatter

[CFDateFormatterCopyProperty](#) (page 6)

Returns a copy of a date formatter's value for a given key.

[CFDateFormatterGetDateStyle](#) (page 11)

Returns the date style used to create the given date formatter object.

[CFDateFormatterGetFormat](#) (page 11)

Returns a format string for the given date formatter object.

[CFDateFormatterGetLocale](#) (page 11)

Returns the locale object used to create the given date formatter object.

[CFDateFormatterGetTimeStyle](#) (page 12)

Returns the time style used to create the given date formatter object.

## Getting the CFDateFormatter Type ID

[CFDateFormatterGetTypeID](#) (page 12)

Returns the type identifier for CFDateFormatter.

# Functions

### **CFDateFormatterCopyProperty**

Returns a copy of a date formatter's value for a given key.

```

CTypeRef CFDateFormatterCopyProperty (
    CFDateFormatterRef formatter,
    CFStringRef key
);

```

**Parameters***formatter*

The date formatter to examine.

*key*

The property key for the value to obtain. See [“Date Formatter Property Keys”](#) (page 16) for a description of possible values for this parameter.

**Return Value**

A CType object that is a copy of the property value for *key*, or NULL if there is no value specified for *key*. Ownership follows the Create Rule.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterCreate**

Creates a new CFDateFormatter object, localized to the given locale, which will format dates to the given date and time styles.

```

CFDateFormatterRef CFDateFormatterCreate (
    CFAllocatorRef allocator,
    CFLocaleRef locale,
    CFDateFormatterStyle dateStyle,
    CFDateFormatterStyle timeStyle
);

```

**Parameters***alloc*

The allocator to use to allocate memory for the new object. Pass NULL or `kCFAllocatorDefault` to use the current default allocator.

*locale*

The locale to use for localization. If NULL uses the default system local. Use `CFLocaleCopyCurrent` to specify the locale of the current user.

*dateStyle*

The date style to use when formatting dates. See [“Date Formatter Styles”](#) (page 14) for possible values.

*timeStyle*

The time style to use when formatting times. See [“Date Formatter Styles”](#) (page 14) for possible values.

**Return Value**

A new date formatter, localized to the given locale, which will format dates to the given date and time styles. Returns NULL if there was a problem creating the object. Ownership follows the Create Rule.

**Discussion**

You can use `kCFDateFormatterNoStyle` to suppress output for the date or time. The following code fragment illustrates the creation and use of a date formatter that only outputs the date information (memory management is omitted for clarity).

```
CFLocaleRef locale = CFLocaleCreate(kCFAllocatorDefault, CFSTR("en_GB"));

CFDateFormatterRef formatter = CFDateFormatterCreate(
    kCFAllocatorDefault, locale, kCFDateFormatterMediumStyle,
    kCFDateFormatterNoStyle);

CFDateRef date = CFDateCreate(kCFAllocatorDefault, 123456);
CFStringRef dateAsString = CFDateFormatterCreateStringWithDate (
    kCFAllocatorDefault, formatter, date);

CFShow(dateAsString);
// outputs "2 Jan 2001"
```

**Availability**

Available in Mac OS X v10.3 and later.

**Related Sample Code**

CFFTPSample

CFPrefTopScores

**Declared In**

CFDateFormatter.h

**CFDateFormatterCreateDateFromString**

Returns a date object representing a given string.

```
CFDateRef CFDateFormatterCreateDateFromString (
    CFAllocatorRef allocator,
    CFDateFormatterRef formatter,
    CFStringRef string,
    CFRange *rangep
);
```

**Parameters**

*alloc*

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

*formatter*

The date formatter object to use to parse *string*.

*string*

The string that contains the date.

*rangep*

A reference to the range within the string specifying the substring to be parsed. If `NULL`, the whole string is parsed. Upon return, contains the range that defines the extent of the parse (may be less than the given range).



**Return Value**

A new date that represents *string*, or NULL if there was a problem creating the object. Ownership follows the Create Rule.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterCreateStringWithAbsoluteTime**

Returns a string representation of the given absolute time using the specified date formatter.

```
CFStringRef CFDateFormatterCreateStringWithAbsoluteTime (
    CFAllocatorRef allocator,
    CFDateFormatterRef formatter,
    CFAbsoluteTime at
);
```

**Parameters**

*alloc*

The allocator to use to allocate memory for the new object. Pass NULL or `kCFAllocatorDefault` to use the current default allocator.

*formatter*

The date formatter object that specifies the format of the returned string.

*at*

The absolute time for which to generate a string representation.

**Return Value**

A new string that represents *at* in the specified format. Returns NULL if there was a problem creating the object. Ownership follows the Create Rule.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterCreateStringWithDate**

Returns a string representation of the given date using the specified date formatter.

```
CFStringRef CFDateFormatterCreateStringWithDate (
    CFAllocatorRef allocator,
    CFDateFormatterRef formatter,
    CFDateRef date
);
```

**Parameters**

*alloc*

The allocator to use to allocate memory for the new object. Pass NULL or `kCFAllocatorDefault` to use the current default allocator.

*formatter*

The date formatter object that specifies the format of the returned string.

*date*

The date object for which to create a string representation.

#### Return Value

A new string that represents *date* in the specified format. Returns `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

#### Availability

Available in Mac OS X v10.3 and later.

#### Related Sample Code

CFFTPSample

CFPrefTopScores

#### Declared In

CFDateFormatter.h

## CFDateFormatterGetAbsoluteTimeFromString

Returns an absolute time object representing a given string.

```
Boolean CFDateFormatterGetAbsoluteTimeFromString (
    CFDateFormatterRef formatter,
    CFStringRef string,
    CFRange *rangep,
    CFAbsoluteTime *atp
);
```

#### Parameters

*formatter*

The date formatter object to use to parse *string*.

*string*

The string that contains the time to be parsed.

*rangep*

Reference to the range within the string specifying the substring to be parsed. If `NULL`, the whole string is parsed. On return, the range that defines the extent of the parse (may be less than the given range).

*atp*

An absolute time value, returned by reference, that represents *string*. Ownership follows the Get Rule.

#### Return Value

`true` if the string was parsed successfully, otherwise `false`.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

CFDateFormatter.h

## CFDateFormatterGetDateStyle

Returns the date style used to create the given date formatter object.

```
CFDateFormatterStyle CFDateFormatterGetDateStyle (
    CFDateFormatterRef formatter
);
```

### Parameters

*formatter*

The date formatter to examine.

### Return Value

The date style used to create *formatter*.

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

CFDateFormatter.h

## CFDateFormatterGetFormat

Returns a format string for the given date formatter object.

```
CFStringRef CFDateFormatterGetFormat (
    CFDateFormatterRef formatter
);
```

### Parameters

*formatter*

The date formatter to examine.

### Return Value

The format string for *formatter* as was specified by calling the [CFDateFormatterSetFormat](#) (page 12) function, or derived from the date formatter's date or time styles. Ownership follows the Get Rule.

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

CFDateFormatter.h

## CFDateFormatterGetLocale

Returns the locale object used to create the given date formatter object.

```
CFLocaleRef CFDateFormatterGetLocale (
    CFDateFormatterRef formatter
);
```

### Parameters

*formatter*

The date formatter object to examine.

**Return Value**

The locale object used to create *formatter*. Ownership follows the Get Rule.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterGetTimeStyle**

Returns the time style used to create the given date formatter object.

```
CFDateFormatterStyle CFDateFormatterGetTimeStyle (
    CFDateFormatterRef formatter
);
```

**Parameters**

*formatter*

The date formatter to examine.

**Return Value**

The time style used to create *formatter*.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterGetTypeID**

Returns the type identifier for CFDateFormatter.

```
CTypeID CFDateFormatterGetTypeID (
    void
);
```

**Return Value**

The type identifier for the CFDateFormatter opaque type.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterSetFormat**

Sets the format string of the given date formatter to the specified value.

```
void CFDateFormatterSetFormat (
    CFDateFormatterRef formatter,
    CFStringRef formatString
);
```

**Parameters***formatter*

The date formatter to modify.

*formatString*The format string for *formatter*. The syntax of this string is defined by [Unicode Technical Standard #35](#).**Discussion**

The format string may override other properties previously set using other functions. If this function is not called, the default value of the format string is derived from the date formatter's date and time styles.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

**CFDateFormatterSetProperty**

Sets a date formatter property using a key-value pair.

```
void CFDateFormatterSetProperty (
    CFDateFormatterRef formatter,
    CFStringRef key,
    CTypeRef value
);
```

**Parameters***formatter*

The date formatter to modify.

*key*The name of the property to set. See ["Date Formatter Property Keys"](#) (page 16) for a description of possible values for this parameter.*value*The value for *key*. This should be a CType object corresponding to the specified key.**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

CFDateFormatter.h

## Data Types

### CFDateFormatterRef

A reference to a CFDateFormatter object.

```
typedef struct __CFDateFormatter *CFDateFormatterRef;
```

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

CFDateFormatter.h

### CFDateFormatterStyle

Data type for predefined date and time format styles.

```
typedef CFIndex CFDateFormatterStyle;
```

#### Discussion

For possible values, see [“Date Formatter Styles”](#) (page 14).

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

CFDateFormatter.h

## Constants

### Date Formatter Styles

Predefined date and time format styles.

```
enum {
    kCFDateFormatterNoStyle = 0,
    kCFDateFormatterShortStyle = 1,
    kCFDateFormatterMediumStyle = 2,
    kCFDateFormatterLongStyle = 3,
    kCFDateFormatterFullStyle = 4
};
```

**Constants**

`kCFDateFormatterNoStyle`

Specifies no output.

You use this constant to suppress output for the date or time (see [CFDateFormatterCreate](#) (page 7) for more details).

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterShortStyle`

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

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterMediumStyle`

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

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterLongStyle`

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

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterFullStyle`

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

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.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, for example if you are parsing an external data file which contains date information in a fixed format. There are several different "lengths" of the formats:

- "long" era names, for example "Anno Domini" instead of "AD"
- "very short" names for months and weekdays; for example, "F" instead of "Friday"
- "standalone" names for months and weekdays (for some locales or languages, a month name displayed in isolation needs to be written differently than a month name within a displayed date)
- names of quarters; for example, "Q2" for a short quarter name

**Declared In**

`CFDateFormatter.h`

## Date Formatter Property Keys

Keys used in key-value pairs to discover and specify the value of date formatter properties—used in conjunction with [CFDateFormatterCopyProperty](#) (page 6) and [CFDateFormatterSetProperty](#) (page 13).

```
const CFStringRef kCFDateFormatterIsLenient;
const CFStringRef kCFDateFormatterTimeZone;
const CFStringRef kCFDateFormatterCalendarName;
const CFStringRef kCFDateFormatterDefaultFormat;

const CFStringRef kCFDateFormatterTwoDigitStartDate;
const CFStringRef kCFDateFormatterDefaultDate;
const CFStringRef kCFDateFormatterCalendar;
const CFStringRef kCFDateFormatterEraSymbols;
const CFStringRef kCFDateFormatterMonthSymbols;
const CFStringRef kCFDateFormatterShortMonthSymbols;
const CFStringRef kCFDateFormatterWeekdaySymbols;
const CFStringRef kCFDateFormatterShortWeekdaySymbols;
const CFStringRef kCFDateFormatterAMSymbol;
const CFStringRef kCFDateFormatterPMSymbol;

const CFStringRef kCFDateFormatterLongEraSymbols;
const CFStringRef kCFDateFormatterVeryShortMonthSymbols;
const CFStringRef kCFDateFormatterStandaloneMonthSymbols;
const CFStringRef kCFDateFormatterShortStandaloneMonthSymbols;
const CFStringRef kCFDateFormatterVeryShortStandaloneMonthSymbols;
const CFStringRef kCFDateFormatterVeryShortWeekdaySymbols;
const CFStringRef kCFDateFormatterStandaloneWeekdaySymbols;
const CFStringRef kCFDateFormatterShortStandaloneWeekdaySymbols;
const CFStringRef kCFDateFormatterVeryShortStandaloneWeekdaySymbols;
const CFStringRef kCFDateFormatterQuarterSymbols;
const CFStringRef kCFDateFormatterShortQuarterSymbols;
const CFStringRef kCFDateFormatterStandaloneQuarterSymbols;
const CFStringRef kCFDateFormatterShortStandaloneQuarterSymbols;
const CFStringRef kCFDateFormatterGregorianStartDate;
```

### Constants

`kCFDateFormatterIsLenient`

Specifies the lenient property, a `CFBoolean` object where a true value indicates that the parsing of strings into date or absolute time values will be fuzzy.

The formatter will use heuristics to guess at the date which is intended by the string. As with any guessing, it may get the result date wrong (that is, a date other than that which was intended).

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterTimeZone`

Specifies the time zone property, a `CFTimeZone` object.

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.



`kCFDateFormatterCalendarName`

Specifies the calendar name, a CFString object.

With Mac OS X version 10.3, `kCFGregorianCalendar` (page 19) is the only possible value. With Mac OS X version 10.4, `kCFGregorianCalendar` and other calendar names are specified by CFLocale.

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterDefaultFormat`

The original format string for the formatter (given the date & time style and locale specified at creation).

Available in Mac OS X v10.3 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterTwoDigitStartDate`

Specifies the property representing the date from which two-digit years start, a CFDate object.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterDefaultDate`

Specifies the default date property, a CFDate object.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterCalendar`

Specifies the calendar property, a CFCalendar object.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterEraSymbols`

Specifies the era symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterMonthSymbols`

Specifies the month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterShortMonthSymbols`

Specifies the short month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterWeekdaySymbols`

Specifies the weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kCFDateFormatterShortWeekdaySymbols`

Specifies the short weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.4 and later.

Declared in `CFDateFormatter.h`.

`kNSDateFormatterAMSymbol`

Specifies the AM symbol property, a CFString object.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterPMSymbol`

Specifies the PM symbol property, a CFString object.

Available in Mac OS X v10.4 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterLongEraSymbols`

Specifies the long era symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterVeryShortMonthSymbols`

Specifies the very short month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterStandaloneMonthSymbols`

Specifies the standalone month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterShortStandaloneMonthSymbols`

Specifies the short standalone month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterVeryShortStandaloneMonthSymbols`

Specifies the very short standalone month symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterVeryShortWeekdaySymbols`

Specifies the very short weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterStandaloneWeekdaySymbols`

Specifies the standalone weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterShortStandaloneWeekdaySymbols`

Specifies the short standalone weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterVeryShortStandaloneWeekdaySymbols`

Specifies the very short standalone weekday symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterQuarterSymbols`

Specifies the quarter symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterShortQuarterSymbols`

Specifies the short quarter symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterStandaloneQuarterSymbols`

Specifies the standalone quarter symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterShortStandaloneQuarterSymbols`

Specifies the short standalone quarter symbols property, a CFArray of CFString objects.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

`kNSDateFormatterGregorianStartDate`

Specifies the Gregorian start date property, a NSDate object.

This is used to specify the start date for the Gregorian calendar switch from the Julian calendar. Different locales switched at different times. Normally you should just accept the locale's default date for the switch.

Available in Mac OS X v10.5 and later.

Declared in `NSDateFormatter.h`.

### Discussion

The values for these keys are all CFType objects. The specific types for each key are specified above.

### Declared In

`NSDateFormatter.h`

## Calendar Names

Calendar names used by NSDateFormatter.

```
const CFStringRef kCFGregorianCalendar;
```

### Constants

`kCFGregorianCalendar`

The name of the calendar currently supported by the `kNSDateFormatterCalendarName` (page 17) property.

Available in Mac OS X v10.3 and later.

Declared in `CFLocale.h`.

**Declared In**

NSDateFormatter.h

# Document Revision History

---

This table describes the changes to *CFDateFormatter Reference*.

Date	Notes
2007-05-23	Updated to include new API introduced in Mac OS X v10.5.
2006-11-07	Clarified use of <code>kCFDateFormatterNoStyle</code> in <code>CFDateFormatterCreate</code> .
2006-02-07	Fixed link to ICU library.
2005-12-06	Made minor changes to text to conform to reference consistency guidelines.
2005-08-11	Updated URL for ICU library.
2005-04-29	Added new date formatter property keys for Mac OS X 10.4.
2004-04-22	Added links to ICU format string reference.
2003-07-01	First version of this document.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

### Calendar Names 19

CFDateFormatterCopyProperty **function** 6  
CFDateFormatterCreate **function** 7  
CFDateFormatterCreateDateFromString **function** 8  
CFDateFormatterCreateStringWithAbsoluteTime **function** 9  
CFDateFormatterCreateStringWithDate **function** 9  
CFDateFormatterGetAbsoluteTimeFromString **function** 10  
CFDateFormatterGetDateStyle **function** 11  
CFDateFormatterGetFormat **function** 11  
CFDateFormatterGetLocale **function** 11  
CFDateFormatterGetTimeStyle **function** 12  
CFDateFormatterGetTypeID **function** 12  
CFDateFormatterRef **data type** 14  
CFDateFormatterSetFormat **function** 12  
CFDateFormatterSetProperty **function** 13  
CFDateFormatterStyle **data type** 14

## D

---

### Date Formatter Property Keys 16

### Date Formatter Styles 14

## K

---

kCFDateFormatterAMSymbol **constant** 18  
kCFDateFormatterCalendar **constant** 17  
kCFDateFormatterCalendarName **constant** 17  
kCFDateFormatterDefaultDate **constant** 17  
kCFDateFormatterDefaultFormat **constant** 17  
kCFDateFormatterEraSymbols **constant** 17  
kCFDateFormatterFullStyle **constant** 15  
kCFDateFormatterGregorianStartDate **constant** 19  
kCFDateFormatterIsLenient **constant** 16  
kCFDateFormatterLongEraSymbols **constant** 18

kCFDateFormatterLongStyle **constant** 15  
kCFDateFormatterMediumStyle **constant** 15  
kCFDateFormatterMonthSymbols **constant** 17  
kCFDateFormatterNoStyle **constant** 15  
kCFDateFormatterPMSymbol **constant** 18  
kCFDateFormatterQuarterSymbols **constant** 19  
kCFDateFormatterShortMonthSymbols **constant** 17  
kCFDateFormatterShortQuarterSymbols **constant** 19  
kCFDateFormatterShortStandaloneMonthSymbols **constant** 18  
kCFDateFormatterShortStandaloneQuarterSymbols **constant** 19  
kCFDateFormatterShortStandaloneWeekdaySymbols **constant** 18  
kCFDateFormatterShortStyle **constant** 15  
kCFDateFormatterShortWeekdaySymbols **constant** 17  
kCFDateFormatterStandaloneMonthSymbols **constant** 18  
kCFDateFormatterStandaloneQuarterSymbols **constant** 19  
kCFDateFormatterStandaloneWeekdaySymbols **constant** 18  
kCFDateFormatterTimeZone **constant** 16  
kCFDateFormatterTwoDigitStartDate **constant** 17  
kCFDateFormatterVeryShortMonthSymbols **constant** 18  
kCFDateFormatterVeryShortStandaloneMonthSymbols **constant** 18  
kCFDateFormatterVeryShortStandaloneWeekdaySymbols **constant** 19  
kCFDateFormatterVeryShortWeekdaySymbols **constant** 18  
kCFDateFormatterWeekdaySymbols **constant** 17  
kCFGregorianCalendar **constant** 19