CFCalendar Reference

Data Management: Dates, Times, & Numbers

2009-02-04

Apple Inc.
© 2004, 2009 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, iPhone, Leopard, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Numbers is a trademark of Apple Inc.

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

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

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

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Tables

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

CFCalendar Reference

Derived From:	CFType
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFCalendar.h
Companion guides	Locales Programming Guide Date and Time Programming Guide for Core Foundation Internationalization Programming Topics

Overview

The CFCalendar opaque type represents a calendar system. The associated API provides information about a calendar and supports calendrical computations such as determining the range of a given calendrical unit and adding units to a given absolute time.

CFAbsoluteTime is the operational lingua franca of CFCalendar—to do calendar arithmetic, you start and end with an absolute time; to convert between a decomposed date in one calendar and another calendar, you first convert to an absolute time. CFAbsoluteTime provides the absolute scale and epoch for dates and times, which can then be rendered into a particular calendar, for calendrical computations or user display.

In a calendar, day, week, weekday, month, and year numbers are generally 1-based, but there may be calendar-specific exceptions. Ordinal numbers, where they occur, are 1-based. Some calendars represented by this API may have to map their basic unit concepts into year/month/week/day/… nomenclature. For example, a calendar composed of 4 quarters in a year instead of 12 months uses the "month" unit to represent quarters. The particular values of the unit are defined by each calendar, and are not necessarily "consistent with" or have a "correspondence with," values for that unit in another calendar.

Several CFCalendar functions (`CFCalendarComposeAbsoluteTime` (page 11), `CFCalendarDecomposeAbsoluteTime` (page 14), `CFCalendarAddComponents` (page 10), and `CFCalendarGetComponentDifference` (page 15)) take a description string that describes the calendrical components provided in a varargs parameter area. You can provide as many components as you need (or choose to), in whatever order you choose. When there is incomplete information to compute an absolute time, default values similar to 0 and 1 are usually chosen by a calendar, but this is a calendar-specific choice. If you provide inconsistent information, calendar-specific disambiguation is performed (which may involve ignoring one or more of the parameters). The characters of the description string specify the units and order of the parameters which follow. The characters are adopted from the corresponding format characters used by CFDateFormatter when possible, as shown in Table 1.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Table 1 Calendrical components parameter descriptors

Symbol	Meaning	Value Type
y	year	int
M	month	int
d	day	int
H	hour	int
m	minute	int
s	second	int

Information related to formatting dates and times and name-related calendar information is managed by CFDateFormatter.

CFCalendar is subject to some limitations. There is no leap second handling—the existence of leap seconds is ignored as in the other CoreFoundation API. In general, historical accuracy of calendars is not guaranteed. There is currently no API for defining your own calendars.

CFCalendar is "toll-free bridged" with its Cocoa Foundation counterpart, NSCalendar. 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 `NSCalendar *` parameter, you can pass in a `CFCalendarRef`, and in a function where you see a `CFCalendarRef` parameter, you can pass in an NSCalendar instance. See Interchangeable Data Types for more information on toll-free bridging.

The CFCalendar opaque type is available in Mac OS X v10.4 and later.

Functions by Task

Creating a Calendar

`CFCalendarCopyCurrent` (page 12)
: Returns a copy of the logical calendar for the current user.

`CFCalendarCreateWithIdentifier` (page 13)
: Returns a calendar object for the calendar identified by a calendar identifier.

Calendrical Calculations

`CFCalendarAddComponents` (page 10)
: Computes the absolute time when specified components are added to a given absolute time.

`CFCalendarComposeAbsoluteTime` (page 11)
: Computes the absolute time from components in a description string.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

`CFCalendarDecomposeAbsoluteTime` (page 14)
Computes the components which are indicated by the componentDesc description string for the given absolute time.

`CFCalendarGetComponentDifference` (page 15)
Computes the difference between the two absolute times, in terms of specified calendrical components.

Getting Ranges of Units

`CFCalendarGetRangeOfUnit` (page 19)
Returns the range of values that one unit can take on within a larger unit during which a specific absolute time occurs.

`CFCalendarGetOrdinalityOfUnit` (page 18)
Returns the ordinal number of a calendrical unit within a larger unit at a specified absolute time.

`CFCalendarGetTimeRangeOfUnit` (page 19)
Returns by reference the start time and duration of a given calendar unit that contains a given absolute time.

`CFCalendarGetMaximumRangeOfUnit` (page 16)
Returns the maximum range limits of the values that a specified unit can take on in a given calendar.

`CFCalendarGetMinimumRangeOfUnit` (page 17)
Returns the minimum range limits of the values that a specified unit can take on in a given calendar.

Getting and Setting the Time Zone

`CFCalendarCopyTimeZone` (page 13)
Returns a time zone object for a specified calendar.

`CFCalendarSetTimeZone` (page 21)
Sets the time zone for a calendar.

Getting the Identifier

`CFCalendarGetIdentifier` (page 16)
Returns the given calendar's identifier.

Getting and Setting the Locale

`CFCalendarCopyLocale` (page 12)
Returns a locale object for a specified calendar.

`CFCalendarSetLocale` (page 21)
Sets the locale for a calendar.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Getting and Setting Day Information

`CFCalendarGetFirstWeekday` (page 16)
Returns the index of first weekday for a specified calendar.

`CFCalendarSetFirstWeekday` (page 20)
Sets the first weekday for a calendar.

`CFCalendarGetMinimumDaysInFirstWeek` (page 17)
Returns the minimum number of days in the first week of a specified calendar.

`CFCalendarSetMinimumDaysInFirstWeek` (page 21)
Sets the minimum number of days in the first week of a specified calendar.

Getting the Type ID

`CFCalendarGetTypeID` (page 20)
Returns the type identifier for the CFCalendar opaque type.

Functions

CFCalendarAddComponents

Computes the absolute time when specified components are added to a given absolute time.

```
Boolean CFCalendarAddComponents (
   CFCalendarRef calendar,
   CFAbsoluteTime *at,
   CFOptionFlags options,
   const unsigned char *componentDesc,
   ...
);
```

Parameters

calendar
The calendar to use for the computation.

at
A reference to an absolute time. On input, points to the absolute time to which components are to be added; on output, points to the result of the computation.

options
Options for the calculation. For valid values, see "Constants" (page 22).

componentDesc
A string that describes the components provided in the vararg parameters.

...
Vararg parameters giving amounts of each calendrical component in the order specified by *componentDesc*. The amounts to add may be negative, zero, positive, or any combination thereof.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Return Value
TRUE—and in *at* the computed time—if *at* falls inside the defined range of the calendar and it is possible to calculate the absolute time when the components (the calendrical components specified by *componentDesc* and given in the varargs) are added to the input absolute time *at*; otherwise FALSE.

Discussion
Some operations can be ambiguous, and the behavior of the computation is calendar-specific, but generally components are added in the order specified.

If you specify a "wrap" option (kCFCalendarComponentsWrap (page 24)), the specified components should be incremented and wrap around to zero/one on overflow, but should not cause higher units to be incremented. When "Wrap" is false, overflow in a unit carries into the higher units, as in typical addition.

Note that some computations can take a relatively long time to perform.

The following example shows how to add 2 months and 3 days to absolute time at's current value using an existing calendar (gregorian):

```
CFCalendarAddComponents(gregorian, &at, 0, "Md",  2, 3);
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
CFCalendar.h

CFCalendarComposeAbsoluteTime

Computes the absolute time from components in a description string.

```
Boolean CFCalendarComposeAbsoluteTime (
   CFCalendarRef calendar,
   CFAbsoluteTime *at,
   const unsigned char *componentDesc,
   ...
);
```

Parameters

calendar
The calendar to use for the computation.

at
Upon return, contains the computed absolute time.

componentDesc
A string that describes the components provided in the vararg parameters.

...
Vararg parameters giving amounts of each calendrical component in the order specified by *componentDesc*. The amounts to add may be negative, zero, positive, or any combination thereof.

Return Value
TRUE—and in *at* the absolute time computed from the given components—if the *componentDesc* description string can be converted into an absolute time, otherwise FALSE. Also returns FALSE for out-of-range values.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Discussion
When there are insufficient components provided to completely specify an absolute time, a calendar uses default values of its choice. When there is inconsistent information, a calendar may ignore some of the parameters or the function may return `FALSE`. Unnecessary components are ignored (for example, Day takes precedence over Weekday + Weekday ordinal). Note that some computations can take a relatively long time to perform.

The following example shows how to use this function to initialize an absolute time, `at`, to 6 January 1965 14:10:00, for a given calendar `gregorian`.

```
CFCalendarComposeAbsoluteTime(gregorian, &at, "yMdHms",  1965, 1, 6, 14, 10,
00);
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarCopyCurrent

Returns a copy of the logical calendar for the current user.

```
CFCalendarRef CFCalendarCopyCurrent (
   void
);
```

Return Value
The logical calendar for the current user that is formed from the settings for the current user's chosen system locale overlaid with any custom settings the user has specified in System Preferences. This function may return a retained cached object, not a new object. Ownership follows the Create Rule.

Discussion
Settings you get from this calendar do not change if user defaults change so that your operations are consistent.

Typically you perform some operations on the returned object and then release it. The returned object may be cached, so you do not need to hold on to it indefinitely.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarCopyLocale

Returns a locale object for a specified calendar.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

```
CFLocaleRef CFCalendarCopyLocale (
   CFCalendarRef calendar
);
```

Parameters

calendar

The calendar to examine.

Return Value

A copy of the locale object for the specified calendar. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFCalendar.h`

CFCalendarCopyTimeZone

Returns a time zone object for a specified calendar.

```
CFTimeZoneRef CFCalendarCopyTimeZone (
   CFCalendarRef calendar
);
```

Parameters

calendar

The calendar to examine.

Return Value

A copy of the time zone object for the specified calendar. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFCalendar.h`

CFCalendarCreateWithIdentifier

Returns a calendar object for the calendar identified by a calendar identifier.

```
CFCalendarRef CFCalendarCreateWithIdentifier (
   CFAllocatorRef allocator,
   CFStringRef identifier
);
```

Parameters

alloc

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

ident

A calendar identifier. Calendar identifier constants are given in `CFLocaleRef`.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Return Value
A calendar object for the calendar identified by *ident*. If the identifier is unknown (if, for example, it is either an unrecognized string, or the calendar is not supported by the current version of the operating system), returns `NULL`. Ownership follows the Create Rule.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarDecomposeAbsoluteTime

Computes the components which are indicated by the componentDesc description string for the given absolute time.

```
Boolean CFCalendarDecomposeAbsoluteTime (
   CFCalendarRef calendar,
   CFAbsoluteTime at,
   const unsigned char *componentDesc,
   ...
);
```

Parameters

calendar
The calendar to use for the computation.

at
An absolute time.

componentDesc
A string that describes the components provided in the vararg parameters.

...
Vararg pointers to storage for each of the desired components. On successful return, the pointers are filled with values of the corresponding components. The type of all units is `int`.

Return Value
`TRUE` if the function is able to compute the components indicated by the *componentDesc* description string for the given absolute time, and fills the values to the components given in the varargs. Returns `FALSE` if the absolute time falls outside the defined range of the calendar, or the computation cannot be performed.

Discussion
The Weekday ordinality, when requested, refers to the next larger (than Week) of the requested units. Some computations can take a relatively long time to perform.

The following example shows how to use this function to determine the current year, month, and day, using an existing calendar (`gregorian`):

```
CFCalendarDecomposeAbsoluteTime(gregorian, CFAbsoluteTimeGetCurrent(), "yMd",
 &year, &month, &day);
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

CFCalendarGetComponentDifference

Computes the difference between the two absolute times, in terms of specified calendrical components.

```
Boolean CFCalendarGetComponentDifference (
   CFCalendarRef calendar,
   CFAbsoluteTime startingAT,
   CFAbsoluteTime resultAT,
   CFOptionFlags options,
   const unsigned char *componentDesc,
   ...
);
```

Parameters

calendar

The calendar to use for the computation.

startingAT

The starting absolute time.

resultAT

The result absolute time.

options

Options for the calculation. For valid values, see "Constants" (page 22).

componentDesc

A string that describes the components provided in the vararg parameters.

...

Vararg pointers to storage for each of the desired components. On successful return, the pointers are filled with values of the corresponding components. The type of all units is `int`.

Return Value

`TRUE`—and in the varargs the differences—if it is possible to calculate the difference (result - starting) between *resultAT* and *startingAT* in terms of the calendrical components specified by *componentDesc*. Returns `FALSE` if either absolute time falls outside the defined range of the calendar, or the computation cannot be performed.

Discussion

The result is lossy if there isn't a small enough unit requested to hold the full precision of the difference. Some operations can be ambiguous, and the behavior of the computation is calendar-specific, but generally larger components will be computed before smaller components; for example, in the Gregorian calendar a result might be `1` month and `5` days, instead of, for example, `0` months and `35` days. The resulting component values may be negative if later is before earlier.

This computation is roughly the inverse of the `CFCalendarAddComponents` (page 10) operation, but calendrical arithmetic is invertible only in simple cases. This computation tends to be several times more expensive than the Add operation.

The following example shows how to get the approximate number of days between two absolute times (`at1`, `at2`) using an existing calendar (`gregorian`):

```
CFCalendarGetComponentDifference(gregorian, at1, at2, 0, "d",  &days);
```

Availability

Available in Mac OS X v10.4 and later.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Declared In
`CFCalendar.h`

CFCalendarGetFirstWeekday

Returns the index of first weekday for a specified calendar.

```
CFIndex CFCalendarGetFirstWeekday (
   CFCalendarRef calendar
);
```

Parameters

calendar
The calendar to examine.

Return Value
The index of the first weekday of the specified calendar.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarGetIdentifier

Returns the given calendar's identifier.

```
CFStringRef CFCalendarGetIdentifier (
   CFCalendarRef calendar
);
```

Parameters

calendar
The calendar to examine.

Return Value
A string representation of *calendar*'s identifier. Calendar identifier constants can be found in `CFLocaleRef`. Ownership follows the Get Rule.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarGetMaximumRangeOfUnit

Returns the maximum range limits of the values that a specified unit can take on in a given calendar.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

```
CFRange CFCalendarGetMaximumRangeOfUnit (
   CFCalendarRef calendar,
   CFCalendarUnit unit
);
```

Parameters

calendar

The calendar to examine.

unit

A calendar unit. For valid values see `CFCalendarUnit` (page 22).

Return Value

The maximum range limits of the values that the specified unit can take on in *calendar*. For example, in the Gregorian calendar the maximum ranges for the Day unit is 1-31.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFCalendar.h`

CFCalendarGetMinimumDaysInFirstWeek

Returns the minimum number of days in the first week of a specified calendar.

```
CFIndex CFCalendarGetMinimumDaysInFirstWeek (
   CFCalendarRef calendar
);
```

Parameters

calendar

The calendar to examine.

Return Value

The minimum number of days in the first week of *calendar*.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFCalendar.h`

CFCalendarGetMinimumRangeOfUnit

Returns the minimum range limits of the values that a specified unit can take on in a given calendar.

```
CFRange CFCalendarGetMinimumRangeOfUnit (
   CFCalendarRef calendar,
   CFCalendarUnit unit
);
```

Parameters

calendar

The calendar to examine.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

unit

A calendar unit. For valid values see CFCalendarUnit (page 22).

Return Value

The minimum range limits of the values that the specified unit can take on in *calendar*. For example, in the Gregorian calendar the minimum ranges for the Day unit is 1-28.

Availability

Available in Mac OS X v10.4 and later.

Declared In

CFCalendar.h

CFCalendarGetOrdinalityOfUnit

Returns the ordinal number of a calendrical unit within a larger unit at a specified absolute time.

```
CFIndex CFCalendarGetOrdinalityOfUnit (
   CFCalendarRef calendar,
   CFCalendarUnit smallerUnit,
   CFCalendarUnit biggerUnit,
   CFAbsoluteTime at
);
```

Parameters

calendar

The calendar to examine.

smallerUnit

A calendar unit. For valid values see CFCalendarUnit (page 22).

biggerUnit

A calendar unit. For valid values see CFCalendarUnit (page 22).

at

An absolute time.

Return Value

The ordinal number of the calendar unit specified by *smallerUnit* within the calendar unit specified by *biggerUnit* at the absolute time *at*. For example, the time 00:45 is in the first hour of the day, and for units Hour and Day respectively, the result would be 1.

If the *biggerUnit* parameter is not logically bigger than the *smallerUnit* parameter in the calendar, or the given combination of units does not make sense (or is a computation which is undefined), the result is kCFNotFound.

Discussion

The ordinality is in most cases not the same as the decomposed value of the unit. Typically return values are 1 and greater; an exception is the week-in-month calculation, which returns 0 for days before the first week in the month containing the date. Note that some computations can take a relatively long time to perform.

Availability

Available in Mac OS X v10.4 and later.

Declared In

CFCalendar.h

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

CFCalendarGetRangeOfUnit

Returns the range of values that one unit can take on within a larger unit during which a specific absolute time occurs.

```
CFRange CFCalendarGetRangeOfUnit (
   CFCalendarRef calendar,
   CFCalendarUnit smallerUnit,
   CFCalendarUnit biggerUnit,
   CFAbsoluteTime at
);
```

Parameters

calendar

The calendar to examine.

smallerUnit

A calendar unit. For valid values see `CFCalendarUnit` (page 22).

biggerUnit

A calendar unit. For valid values see `CFCalendarUnit` (page 22).

at

An absolute time.

Return Value

The range of values that the calendar unit specified by *smallerUnit* can take on within the calendar unit specified by *biggerUnit* that includes the absolute time *at*. For example, the range the Day unit can take on in the Month in which the absolute time lies.

If *biggerUnit* is not logically bigger than *smallerUnit* in the calendar, or the given combination of units does not make sense (or is a computation which is undefined), the result is `{kCFNotFound, kCFNotFound}`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFCalendar.h`

CFCalendarGetTimeRangeOfUnit

Returns by reference the start time and duration of a given calendar unit that contains a given absolute time.

```
Boolean CFCalendarGetTimeRangeOfUnit (
    CFCalendarRef calendar,
    CFCalendarUnit unit,
    CFAbsoluteTime at,
    CFAbsoluteTime *startp,
    CFTimeInterval *tip
);
```

Parameters

calendar

The calendar to examine.

unit

A calendar unit (for valid values, see `CFCalendarUnit` (page 22)).

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

at
An absolute time.

startp
Upon return, contains the beginning of the calendar unit specified by *unit* that contains the time *at*.

tip
Upon return, contains the duration of the calendar unit specified by *unit* that contains the time *at*.

Return Value
`true` if the values of *startp* and *tip* could be calculated, otherwise `false`.

Discussion
The function may fail if, for example, you try to get the range of a `kCFCalendarUnitWeekday` and specify a time (*at*) that is during a weekend.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CFCalendar.h`

CFCalendarGetTypeID

Returns the type identifier for the CFCalendar opaque type.

```
CFTypeID CFCalendarGetTypeID (
   void
);
```

Return Value
The type identifier for the CFCalendar opaque type.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarSetFirstWeekday

Sets the first weekday for a calendar.

```
void CFCalendarSetFirstWeekday (
   CFCalendarRef calendar,
   CFIndex wkdy
);
```

Parameters

calendar
The calendar to modify.

wkdy
The index to set for the first weekday of *calendar*.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarSetLocale

Sets the locale for a calendar.

```
void CFCalendarSetLocale (
   CFCalendarRef calendar,
   CFLocaleRef locale
);
```

Parameters

calendar
The calendar to modify.

locale
The locale to set for *calendar*.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarSetMinimumDaysInFirstWeek

Sets the minimum number of days in the first week of a specified calendar.

```
void CFCalendarSetMinimumDaysInFirstWeek (
   CFCalendarRef calendar,
   CFIndex mwd
);
```

Parameters

calendar
The calendar to modify.

mwd
The number to set as the minimum number of days in the first week of *calendar*.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFCalendar.h`

CFCalendarSetTimeZone

Sets the time zone for a calendar.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

```
void CFCalendarSetTimeZone (
   CFCalendarRef calendar,
   CFTimeZoneRef tz
);
```

Parameters

calendar

The calendar to modify.

locale

The time zone to set for *calendar*.

Availability
Available in Mac OS X v10.4 and later.

Declared In
CFCalendar.h

Data Types

CFCalendarRef

A reference to a CFCalendar object.

```
typedef const struct __CFCalendar *CFCalendarRef;
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
CFCalendar.h

Constants

CFCalendarUnit

CFCalendarUnit constants are used to specify calendrical units, such as day or month, in various calendar calculations.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

```
typedef enum {
    kCFCalendarUnitEra = (1 << 1),
    kCFCalendarUnitYear = (1 << 2),
    kCFCalendarUnitMonth = (1 << 3),
    kCFCalendarUnitDay = (1 << 4),
    kCFCalendarUnitHour = (1 << 5),
    kCFCalendarUnitMinute = (1 << 6),
    kCFCalendarUnitSecond = (1 << 7),
    kCFCalendarUnitWeek = (1 << 8),
    kCFCalendarUnitWeekday = (1 << 9),
    kCFCalendarUnitWeekdayOrdinal = (1 << 10),
} CFCalendarUnit;
```

Constants

`kCFCalendarUnitEra`
Specifies the era unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitYear`
Specifies the year unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitMonth`
Specifies the month unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitDay`
Specifies the day unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitHour`
Specifies the hour unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitMinute`
Specifies the minute unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitSecond`
Specifies the second unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

`kCFCalendarUnitWeek`
Specifies the week unit.

Available in Mac OS X v10.4 and later.

Declared in `CFCalendar.h`.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

`kCFCalendarUnitWeekday`
: Specifies the weekday unit.

  The weekday units are the numbers `1-N` (where for the Gregorian calendar `N=7` and `1` is Sunday).

  Available in Mac OS X v10.4 and later.

  Declared in `CFCalendar.h`.

`kCFCalendarUnitWeekdayOrdinal`
: Specifies the ordinal weekday unit.

  The weekday ordinal unit describes ordinal position within the month unit of the corresponding weekday unit. For example, in the Gregorian calendar a weekday ordinal unit of `2` for a weekday unit `3` indicates "the second Tuesday in the month".

  Available in Mac OS X v10.4 and later.

  Declared in `CFCalendar.h`.

Component Wrapping Options

The wrapping option specifies overflow behavior for calendar components in calendrical calculations—see `CFCalendarAddComponents` (page 10) and `CFCalendarGetComponentDifference` (page 15).

```
enum {
    kCFCalendarComponentsWrap = (1 << 0)
}
```

Constants

`kCFCalendarComponentsWrap`
: Specifies that the components specified for calendar components should be incremented and wrap around to zero/one on overflow, but should not cause higher units to be incremented.

  Available in Mac OS X v10.4 and later.

  Declared in `CFCalendar.h`.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Document Revision History

This table describes the changes to *CFCalendar Reference*.

Date	Notes
2009-02-04	Clarified return value of CFCalendarGetOrdinalityOfUnit.
2006-12-11	Clarified definitions of kCFCalendarUnitWeekday and kCFCalendarUnitWeekdayOrdinal; corrected definition of calendrical unit for seconds.
Leopard WWDC	Corrected minor typographical errors.
2006-05-23	Corrected code examples using CFCalendarDecomposeAbsoluteTime, CFCalendarGetComponentDifference, CFCalendarAddComponents, and CFCalendarComposeAbsoluteTime.
2005-12-06	Made corrections to the Companion Documents list.
2005-11-09	Made corrections in Companion Documents list.
2005-04-29	First version of this document.

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.

Document Revision History

2009-02-04 | © 2004, 2009 Apple Inc. All Rights Reserved.