
Time Utilities Reference

Core Foundation



2007-08-23



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 and the Apple logo are trademarks of Apple Inc., registered in the United States and other countries.

iPhone and Numbers are trademarks of Apple Inc.

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

Time Utilities Reference 5

Overview 5

Functions 5

 CFAbsoluteTimeAddGregorianUnits 5

 CFAbsoluteTimeGetCurrent 6

 CFAbsoluteTimeGetDayOfWeek 6

 CFAbsoluteTimeGetDayOfYear 7

 CFAbsoluteTimeGetDifferenceAsGregorianUnits 7

 CFAbsoluteTimeGetGregorianDate 8

 CFAbsoluteTimeGetWeekOfYear 9

 CFGregorianCalendarGetAbsoluteTime 9

 CFGregorianCalendarIsValid 10

Data Types 11

 CFAbsoluteTime 11

 CFGregorianCalendar 11

 CFGregorianCalendarUnits 11

 CFTimeInterval 12

Constants 12

 CFGregorianCalendarUnitFlags 12

 Predefined Time Interval Values 14

Document Revision History 15

Index 17

Time Utilities Reference

Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Dates and Times Programming Guide for Core Foundation
Declared in	CFDate.h

Overview

Core Foundation measures time in units of seconds. The base data type is the [CFTimeInterval](#) (page 12), which measures the difference in seconds between two times. Fixed times, or dates, are defined by the [CFAbsoluteTime](#) (page 11) data type, which measures the time interval between a particular date and the absolute reference date of Jan 1 2001 00:00:00 GMT.

The [CFGregorianCalendar](#) (page 11) structure represents absolute times in terms of the Gregorian calendar. Functions such as [CFAbsoluteTimeGetGregorianCalendar](#) (page 8) use a [CFTimeZone](#) object to obtain the local time in a particular time zone.

The [CFDate](#) opaque type wraps an absolute time into a CType-base object, allowing you to put time objects into collections and property lists and to be handled by other object-oriented parts of Core Foundation.

Functions

CFAbsoluteTimeAddGregorianUnits

Adds a time interval, expressed as Gregorian units, to a given absolute time.

```
CFAbsoluteTime CFAbsoluteTimeAddGregorianUnits (  
    CFAbsoluteTime at,  
    CFTimeZoneRef tz,  
    CFGregorianCalendarUnits units  
);
```

Parameters

at

The absolute time to which the interval is added.

tz

The time zone to use for time correction. Pass `NULL` for GMT.

units

The time interval to add to *at*.

Return Value

An absolute time value equal to the sum of *at* and *units*.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CalendarView

Declared In

CFDate.h

CFAbsoluteTimeGetCurrent

Returns the current system absolute time.

```
CFAbsoluteTime CFAbsoluteTimeGetCurrent ();
```

Return Value

The current absolute time.

Discussion

Absolute time is measured in seconds relative to the absolute reference date of Jan 1 2001 00:00:00 GMT. A positive value represents a date after the reference date, a negative value represents a date before it. For example, the absolute time `-32940326` is equivalent to December 16th, 1999 at 17:54:34. Repeated calls to this function do not guarantee monotonically increasing results. The system time may decrease due to synchronization with external time references or due to an explicit user change of the clock.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

AudioQueueTools

ColorSyncDevices

NSGLImage

OpenGL Screensaver

Worm

Declared In

CFDate.h

CFAbsoluteTimeGetDayOfWeek

Returns an integer representing the day of the week indicated by the specified absolute time.

```
SInt32 CFAbsoluteTimeGetDayOfWeek (
    CFAbsoluteTime at,
    CFTimeZoneRef tz
);
```

Parameters

at

The absolute time to convert.

tz

The time zone to use for time correction. Pass `NULL` for GMT.

Return Value

An integer (1-7) representing the day of the week specified by *at*. Per ISO-8601, Monday is represented by 1, Tuesday by 2, and so on.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CalendarView

Declared In

CFDate.h

CFAbsoluteTimeGetDayOfYear

Returns an integer representing the day of the year indicated by the specified absolute time.

```
SInt32 CFAbsoluteTimeGetDayOfYear (
    CFAbsoluteTime at,
    CFTimeZoneRef tz
);
```

Parameters

at

The absolute time to convert.

tz

The time zone to use for time correction. Pass `NULL` for GMT.

Return Value

An integer (1-366) representing the day of the year specified by *at*.

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CFAbsoluteTimeGetDifferenceAsGregorianUnits

Computes the time difference between two specified absolute times and returns the result as an interval in Gregorian units.

```

CFGregorianUnits CFAbsoluteTimeGetDifferenceAsGregorianUnits (
    CFAbsoluteTime at1,
    CFAbsoluteTime at2,
    CFTimeZoneRef tz,
    CFOptionFlags unitFlags
);

```

Parameters*at1*

An absolute time.

at2

An absolute time.

*tz*The time zone to use for time correction. Pass `NULL` for GMT.*unitFlags*A mask that specifies which Gregorian unit fields to use when converting the absolute time difference into a Gregorian interval. See [Gregorian Unit Flags](#) (page 12) for a list of values from which to construct the mask.**Return Value**The difference between the specified absolute times (as $at1 - at2$ —if *at1* is earlier than *at2*, the result is negative) expressed in the units specified by *unitFlags*.**Discussion**

The temporal difference is expressed as accurately as possible, given the units specified. For example, if you asked for the number of months and hours between 2:30pm on April 8 2005 and 5:45pm September 9 2005, the result would be 5 months and 27 hours.

The following example prints the number of hours and minutes between the current time (now) and the reference date (1 January 2001 00:00:00 GMT).

```

CFAbsoluteTime now = CFAbsoluteTimeGetCurrent ();

CFGregorianUnits units = CFAbsoluteTimeGetDifferenceAsGregorianUnits
    (now, 0, NULL, (kCFGregorianUnitsHours | kCFGregorianUnitsMinutes));

CFStringRef output = CFStringCreateWithFormat
    (NULL, 0, CFSTR("hours: %d; minutes: %d"), units.hours, units.minutes);
CFShow(output);

```

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CFAbsoluteTimeGetGregorianDate

Converts an absolute time value into a Gregorian date.


```
CFGregorianCalendar CFAbsoluteTimeGetGregorianCalendar (
    CFAbsoluteTime at,
    CFTimeZoneRef tz
);
```

Parameters

at
The absolute time value to convert.

tz
The time zone to use for time correction. Pass NULL for GMT.

Return Value

The GregorianCalendar equivalent for *at*.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CalendarView
SampleDS

Declared In

CFDate.h

CFAbsoluteTimeGetWeekOfYear

Returns an integer representing the week of the year indicated by the specified absolute time.

```
SInt32 CFAbsoluteTimeGetWeekOfYear (
    CFAbsoluteTime at,
    CFTimeZoneRef tz
);
```

Parameters

at
The absolute time to convert.

tz
The time zone to use for time correction. Pass NULL for GMT.

Return Value

An integer (1-53) representing the week of the year specified by *at*. The numbering follows the ISO 8601 definition of week.

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CFGregorianCalendarGetAbsoluteTime

Converts a GregorianCalendar date value into an absolute time value.

```
CFAbsoluteTime CFGregorianCalendarGetAbsoluteTime (
    CFGregorianCalendar gdate,
    CFTimeZoneRef tz
);
```

Parameters*gdate*

The Gregorian date to convert.

*tz*The time zone to use for time correction. Pass `NULL` for GMT.**Return Value**The absolute time equivalent of *gdate*.**Availability**

Available in Mac OS X v10.0 and later.

Related Sample Code

CalendarView

Declared In

CFDate.h

CFGregorianCalendarIsValid

Checks the specified fields of a CFGregorianCalendar structure for valid values.

```
Boolean CFGregorianCalendarIsValid (
    CFGregorianCalendar gdate,
    CFOptionFlags unitFlags
);
```

Parameters*gdate*

The CFGregorianCalendar structure whose fields to validate.

*unitFlags*A mask that specifies which Gregorian unit fields to validate. See [Gregorian Unit Flags](#) (page 12) for a list of values from which to construct the mask.**Return Value**`true` if the specified fields are valid, otherwise `false`.**Availability**

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

Data Types

CFAbsoluteTime

Type used to represent a specific point in time relative to the absolute reference date of 1 Jan 2001 00:00:00 GMT.

```
typedef CTimeInterval CFAbsoluteTime;
```

Discussion

Absolute time is measured by the number of seconds between the reference date and the specified date. Negative values indicate dates/times before the reference date. Positive values indicate dates/times after the reference date.

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CFGregorianCalendar

Structure used to represent a point in time using the Gregorian calendar.

```
struct CFGregorianCalendar {
    SInt32 year;
    SInt8 month;
    SInt8 day;
    SInt8 hour;
    SInt8 minute;
    double second;
};
typedef struct CFGregorianCalendar CFGregorianCalendar;
```

Discussion

CFGregorianCalendar is implemented using the smallest data type appropriate for the range of possible values. For example, there are only 12 months in the Gregorian year, so there is no need to use an integer type larger than 8 bits. To represent a time interval in Gregorian units, use a [CFGregorianCalendarUnits](#) (page 11).

The month and day units are 1-based: the index for January is 1, and the index for the first day of the month is 1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CFGregorianCalendarUnits

Structure used to represent a time interval in Gregorian units.

```

struct CFGregorianUnits {
    SInt32 years;
    SInt32 months;
    SInt32 days;
    SInt32 hours;
    SInt32 minutes;
    double seconds;
};
typedef struct CFGregorianUnits CFGregorianUnits;

```

Discussion

A CFGregorianUnits is used to represent arbitrary time *intervals* (to represent a point in time using Gregorian units, use a [CFGregorianDate](#) (page 11)). Each field can take values up to the maximum possible for its data type. Negative values are also valid.

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

CTimeInterval

Type used to represent elapsed time in seconds.

```
typedef double CFTimeInterval;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFDate.h

Constants

CFGregorianUnitFlags

These option flags are used as a mask to indicate a specific set of fields in the CFGregorianDate or CFGregorianUnits structures.

```
enum CFGregorianUnitFlags {
    kCFGregorianUnitsYears = (1 << 0),
    kCFGregorianUnitsMonths = (1 << 1),
    kCFGregorianUnitsDays = (1 << 2),
    kCFGregorianUnitsHours = (1 << 3),
    kCFGregorianUnitsMinutes = (1 << 4),
    kCFGregorianUnitsSeconds = (1 << 5),
    kCFGregorianAllUnits = 0x00FFFFFF
};
typedef enum CFGregorianUnitFlags CFGregorianUnitFlags;
```

Constants

kCFGregorianUnitsYears

Specifies the year field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianUnitsMonths

Specifies the month field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianUnitsDays

Specifies the day field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianUnitsHours

Specifies the hours field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianUnitsMinutes

Specifies the minutes field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianUnitsSeconds

Specifies the seconds field.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

kCFGregorianAllUnits

Specifies all fields.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

Discussion

These flags are used with functions such as [CFGregorianDateIsValid](#) (page 10) and [CFAbsoluteTimeGetDifferenceAsGregorianUnits](#) (page 7) which operate on a `CFGregorianDate` or `CFGregorianUnits` structure. For more details, see the discussion of those functions.

Declared In

`CFDate.h`

Predefined Time Interval Values

Time intervals between the absolute reference date and certain other dates.

```
const CTimeInterval kCFAbsoluteTimeIntervalSince1970;  
const CTimeInterval kCFAbsoluteTimeIntervalSince1904;
```

Constants

`kCFAbsoluteTimeIntervalSince1970`

The time interval between 1 January 1970 and the reference date 1 January 2001 00:00:00 GMT.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

`kCFAbsoluteTimeIntervalSince1904`

The time interval between 1 January 1904 and the reference date 1 January 2001 00:00:00 GMT.

Available in Mac OS X v10.0 and later.

Declared in `CFDate.h`.

Document Revision History

This table describes the changes to *Time Utilities Reference*.

Date	Notes
2007-08-23	Clarified base index for date units.
2006-05-23	Clarified the definition of <code>CFAbsoluteTimeGetDifferenceAsGregorianUnits</code> .
2005-12-06	Made minor changes to text to conform to reference consistency guidelines.
2005-08-11	Cosmetic changes to conform to documentation guidelines.
2004-08-31	Minor correction to discussion of <code>CFAbsoluteTimeGetCurrent</code> (page 6).
2003-01-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

C

CFAbsoluteTime **data type** [11](#)
CFAbsoluteTimeAddGregorianUnits **function** [5](#)
CFAbsoluteTimeGetCurrent **function** [6](#)
CFAbsoluteTimeGetDayOfWeek **function** [6](#)
CFAbsoluteTimeGetDayOfYear **function** [7](#)
CFAbsoluteTimeGetDifferenceAsGregorianUnits
function [7](#)
CFAbsoluteTimeGetGregorianDate **function** [8](#)
CFAbsoluteTimeGetWeekOfYear **function** [9](#)
CFGregorianDate **structure** [11](#)
CFGregorianDateGetAbsoluteTime **function** [9](#)
CFGregorianDateIsValid **function** [10](#)
CFGregorianUnitFlags **12**
CFGregorianUnits **structure** [11](#)
CFTimeInterval **data type** [12](#)

K

kCFAbsoluteTimeIntervalSince1904 **constant** [14](#)
kCFAbsoluteTimeIntervalSince1970 **constant** [14](#)
kCFGregorianAllUnits **constant** [13](#)
kCFGregorianUnitsDays **constant** [13](#)
kCFGregorianUnitsHours **constant** [13](#)
kCFGregorianUnitsMinutes **constant** [13](#)
kCFGregorianUnitsMonths **constant** [13](#)
kCFGregorianUnitsSeconds **constant** [13](#)
kCFGregorianUnitsYears **constant** [13](#)

P

Predefined Time Interval Values [14](#)