

---

# CFNotificationCenter Reference

Data Management: Event Handling



2007-01-22



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, 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

---

## **CFNotificationCenter Reference 5**

---

Overview	5
Functions by Task	6
Accessing a Notification Center	6
Posting a Notification	6
Adding and Removing Observers	6
Getting the CFNotificationCenter Type ID	6
Functions	6
CFNotificationCenterAddObserver	6
CFNotificationCenterGetDarwinNotifyCenter	8
CFNotificationCenterGetDistributedCenter	8
CFNotificationCenterGetLocalCenter	9
CFNotificationCenterGetTypeID	9
CFNotificationCenterPostNotification	9
CFNotificationCenterPostNotificationWithOptions	10
CFNotificationCenterRemoveEveryObserver	11
CFNotificationCenterRemoveObserver	12
Callbacks	12
CFNotificationCenterCallback	12
Data Types	13
CFNotificationCenterRef	13
Constants	14
Notification Delivery Suspension Behavior	14
Notification Posting Options	15

---

## **Document Revision History 17**

---

## **Index 19**

---



# CFNotificationCenter Reference

---

<b>Derived From:</b>	CFType
<b>Framework:</b>	CoreFoundation/CoreFoundation.h
<b>Companion guide</b>	Notification Programming Topics for Cocoa
<b>Declared in</b>	CFNotificationCenter.h

## Overview

A `CFNotificationCenter` object provides the means by which you can send a message, or notification, to any number of recipients, or observers, without having to know anything about the recipients. A notification message consists of a notification name (a `CFString`), a pointer value that identifies the object posting the notification, and an optional dictionary that contains additional information about the particular notification.

To register as an observer of a notification, you call `CFNotificationCenterAddObserver` (page 6), providing an identifier for your observer, the callback function that should be called when the notification is posted, and the name of the notification and the object in which you are interested. The observer identifier is passed back to the callback function, along with the notification information. You can use the identifier to distinguish multiple observers using the same callback function. The identifier is also used to unregister the observer with `CFNotificationCenterRemoveObserver` (page 12) and `CFNotificationCenterRemoveEveryObserver` (page 11).

To send a notification, you call `CFNotificationCenterPostNotification` (page 9), passing in the notification information. The notification center then looks up all the observers that registered for this notification and sends the notification information to their callback functions.

There are three types of `CFNotificationCenter`—a distributed notification center, a local notification center, and a Darwin notification center—an application may have at most one of each type. The distributed notification is obtained with `CFNotificationCenterGetDistributedCenter` (page 8). A distributed notification center delivers notifications between applications. In this case, the notification object must always be a `CFString` object and the notification dictionary must contain only property list values. The local and Darwin notification centers are available in Mac OS X version 10.4 and later, and obtained using `CFNotificationCenterGetLocalCenter` (page 9) and `CFNotificationCenterGetDarwinNotifyCenter` (page 8) respectively.

Unlike some other Core Foundation opaque types with names similar to a Cocoa Foundation class (such as `CFString` and `NSString`), `CFNotificationCenter` objects cannot be cast (“toll-free bridged”) to `NSNotificationCenter` objects or vice-versa.

## Functions by Task

### Accessing a Notification Center

[CFNotificationCenterGetDarwinNotifyCenter](#) (page 8)

Returns the application's Darwin notification center.

[CFNotificationCenterGetDistributedCenter](#) (page 8)

Returns the application's distributed notification center.

[CFNotificationCenterGetLocalCenter](#) (page 9)

Returns the application's local notification center.

### Posting a Notification

[CFNotificationCenterPostNotification](#) (page 9)

Posts a notification for an object.

[CFNotificationCenterPostNotificationWithOptions](#) (page 10)

Posts a notification for an object using specified options.

### Adding and Removing Observers

[CFNotificationCenterAddObserver](#) (page 6)

Registers an observer to receive notifications.

[CFNotificationCenterRemoveEveryObserver](#) (page 11)

Stops an observer from receiving any notifications from any object.

[CFNotificationCenterRemoveObserver](#) (page 12)

Stops an observer from receiving certain notifications.

### Getting the CFNotificationCenter Type ID

[CFNotificationCenterGetTypeID](#) (page 9)

Returns the type identifier for the CFNotificationCenter opaque type.

## Functions

### CFNotificationCenterAddObserver

Registers an observer to receive notifications.

```
void CFNotificationCenterAddObserver (
    CFNotificationCenterRef center,
    const void *observer,
    CFNotificationCallback callBack,
    CFStringRef name,
    const void *object,
    CFNotificationSuspensionBehavior suspensionBehavior
);
```

**Parameters***center*

The notification center to which to add the observer.

*observer*

The observer. In Mac OS X v10.3 and later, this parameter may be NULL.

*callBack*

The callback function to call when *object* posts the notification named *name*.

*name*

The name of the notification to observe. If NULL, *callBack* is called for any notification posted by *object*.

If *center* is a Darwin notification center, this value must *not* be NULL.

*object*

The object to observe. For distributed notifications, *object* must be a CFString object. If NULL, *callBack* is called when a notification named *name* is posted by any object.

If *center* is a Darwin notification center, this value is ignored.

*suspensionBehavior*

Flag indicating how notifications should be handled when the application is in the background. See [“Notification Delivery Suspension Behavior”](#) (page 14) for the list of available values.

If *center* is a Darwin notification center, this value is ignored.

**Discussion**

The first time an observer is registered with a distributed notification center, the notification center creates a connection with the system-wide notification server and places a listening port into the common modes of the current thread’s run loop. When a notification is delivered, it is processed on this initial thread, even if the observer that is receiving the notification registered for the notification on a different thread.

Because loaded frameworks may potentially spawn threads and add their own observers before your code executes, you cannot know for certain which thread will receive distributed notifications. If you need to control which thread processes a notification, your callback function must be able to forward the notification to the proper thread. You can use a CFMessagePort object or a custom CFRunLoopSource object to send notifications to the correct thread’s run loop.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

NotificationObserver

VideoViewer

XcodeClientServer

**Declared In**

CFNotificationCenter.h

## CFNotificationCenterGetDarwinNotifyCenter

Returns the application's Darwin notification center.

```

CFNotificationCenterRef CFNotificationCenterGetDarwinNotifyCenter (
    void
);

```

### Return Value

The application's Darwin notification center.

### Discussion

This notification center is used to cover the `<notify.h>` Core OS notification mechanism (see `/usr/include/notify.h`). An application has only one Darwin notification center, so this function returns the same value each time it is called.

The Darwin Notify Center has no notion of per-user sessions, all notifications are system-wide. As with distributed notifications, the main thread's run loop must be running in one of the common modes (usually `kCFRunLoopDefaultMode`) for Darwin-style notifications to be delivered.

**Important:** Several function parameters are ignored by Darwin notification centers. To ensure future compatibility, you should pass `NULL` or `0` for all ignored arguments.

### Availability

Available in Mac OS X v10.4 and later.

### Declared In

`CFNotificationCenter.h`

## CFNotificationCenterGetDistributedCenter

Returns the application's distributed notification center.

```

CFNotificationCenterRef CFNotificationCenterGetDistributedCenter (
    void
);

```

### Return Value

The application's distributed notification center. An application has only one distributed notification center, so this function returns the same value each time it is called.

### Discussion

A distributed notification center delivers notifications between applications. A notification object used with a distributed notification center must always be a `CFString` object and the notification dictionary must contain only property list values.

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

[NotificationObserver](#)  
[NotificationPoster](#)  
[VideoViewer](#)  
[XcodeClientServer](#)

**Declared In**

CFNotificationCenter.h

**CFNotificationCenterGetLocalCenter**

Returns the application's local notification center.

```
CFNotificationCenterRef CFNotificationCenterGetLocalCenter (
    void
);
```

**Return Value**

The application's local notification center. An application has only one local notification center, so this function returns the same value each time it is called.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

CFNotificationCenter.h

**CFNotificationCenterGetTypeID**

Returns the type identifier for the CFNotificationCenter opaque type.

```
CTypeID CFNotificationCenterGetTypeID (
    void
);
```

**Return Value**

The type identifier for the CFNotificationCenter opaque type.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CFNotificationCenter.h

**CFNotificationCenterPostNotification**

Posts a notification for an object.

```
void CFNotificationCenterPostNotification (
    CFNotificationCenterRef center,
    CFStringRef name,
    const void *object,
    CFDictionaryRef userInfo,
    Boolean deliverImmediately
);
```

**Parameters***center*

The notification center to post the notification.

*name*

The name of the notification to post. This value must not be `NULL`.

*object*

The object posting the notification. If `NULL`, the notification is sent only to observers that are observing all objects. In other words, only observers that registered for the notification with a `NULL` value for *object* will receive the notification.

If you want to allow your clients to register for notifications using Cocoa APIs (see *NSNotificationCenter Class Reference*), then *object* must be a Core Foundation or Cocoa object.

For distributed notifications, *object* must be a `CFString` object.

If *center* is a Darwin notification center, this value is ignored.

*userInfo*

A dictionary passed to observers. You populate this dictionary with additional information describing the notification. For distributed notifications, the dictionary must contain only property list objects. This value may be `NULL`.

If *center* is a Darwin notification center, this value is ignored.

*deliverImmediately*

If `true`, the notification is delivered to all observers immediately, even if some observers are in suspended (background) applications and they requested different suspension behavior when registering for the notification. If `false`, each observer's requested suspension behavior is respected.

If *center* is a Darwin notification center, this value is ignored.

#### Availability

Available in Mac OS X v10.0 and later.

#### Related Sample Code

NotificationPoster

XcodeClientServer

#### Declared In

CFNotificationCenter.h

## CFNotificationCenterPostNotificationWithOptions

Posts a notification for an object using specified options.

```
void CFNotificationCenterPostNotificationWithOptions (
    CFNotificationCenterRef center,
    CFStringRef name,
    const void *object,
    CFDictionaryRef userInfo,
    CFOptionFlags options
);
```

#### Parameters

*center*

The notification center to post the notification.

*name*

The name of the notification to post. This value must not be `NULL`.

*object*

The object posting the notification. If `NULL`, the notification is sent only to observers that are observing all objects. In other words, only observers that registered for the notification with a `NULL` value for *object* will receive the notification.

If you want to allow your clients to register for notifications using Cocoa APIs (see *NSNotificationCenter Class Reference*), then *object* must be a Core Foundation or Cocoa object.

For distributed notifications, *object* must be a `CFString` object.

If *center* is a Darwin notification center, this value is ignored.

*userInfo*

A dictionary to pass to observers. You populate this dictionary with additional information describing the notification. For distributed notifications, the dictionary must contain only property list objects. Can be `NULL`.

If *center* is a Darwin notification center, this value is ignored.

*options*

Specifies if the notification should be posted immediately, or to all sessions. See “[Notification Posting Options](#)” (page 15) for possible values.

If *center* is a Darwin notification center, this value is ignored.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

`CFNotificationCenter.h`

**CFNotificationCenterRemoveEveryObserver**

Stops an observer from receiving any notifications from any object.

```
void CFNotificationCenterRemoveEveryObserver (
    CFNotificationCenterRef center,
    const void *observer
);
```

**Parameters***center*

The notification center from which to remove observers.

*observer*

The observer. This value must not be `NULL`.

**Discussion**

If you no longer want an observer to receive any notifications, perhaps because the observer is being deallocated, you can call this function to unregister the observer from all the notifications for which it had previously registered.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

`NotificationObserver`

`XcodeClientServer`

**Declared In**

CFNotificationCenter.h

**CFNotificationCenterRemoveObserver**

Stops an observer from receiving certain notifications.

```
void CFNotificationCenterRemoveObserver (
    CFNotificationCenterRef center,
    const void *observer,
    CFStringRef name,
    const void *object
);
```

**Parameters***center*

The notification center to modify.

*observer*

The observer. This value must not be NULL.

*name*The name of the notification to stop observing. If NULL, *observer* stops receiving callbacks for all notifications posted by *object*.*object*The object to stop observing. For distributed notifications, *object* must be a CFString object. If NULL, *observer* stops receiving callbacks for all objects posting notifications named *name*.If *center* is a Darwin notification center, this value is ignored.**Discussion**If both *name* and *object* are NULL, this function unregisters *observer* from all the notifications for which it had previously registered with *center*.**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

NotificationObserver

VideoViewer

XcodeClientServer

**Declared In**

CFNotificationCenter.h

## Callbacks

**CFNotificationCenterCallback**

Callback function invoked for each observer of a notification when the notification is posted.

```
typedef void (*CFNotificationCenterCallback) (
    CFNotificationCenterRef center,
    void *observer,
    CFStringRef name,
    const void *object,
    CFDictionaryRef userInfo
);
```

If you name your function `MyCallback`, you would declare it like this:

```
void MyCallback (
    CFNotificationCenterRef center,
    void *observer,
    CFStringRef name,
    const void *object,
    CFDictionaryRef userInfo
);
```

### Parameters

*center*

The notification center handling the notification.

*observer*

An arbitrary value, other than `NULL`, that identifies the observer.

*name*

The name of the notification being posted.

*object*

An arbitrary value that identifies the object posting the notification. For distributed notifications, *object* is always a `CFString` object. This value could be `NULL`.

*userInfo*

A dictionary containing additional information regarding the notification. This value could be `NULL`.  
If the notification center is a Darwin notification center, this value must be ignored.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

`CFNotificationCenter.h`

## Data Types

### CFNotificationCenterRef

The type of a reference to a `CFNotificationCenter`.

```
typedef struct *CFNotificationCenterRef;
```

### Availability

Available in Mac OS X v10.0 and later.

**Declared In**

CFNotificationCenter.h

## Constants

### Notification Delivery Suspension Behavior

Suspension flags that indicate how distributed notifications should be handled when the receiving application is in the background.

```
enum CFNotificationSuspensionBehavior {
    CFNotificationSuspensionBehaviorDrop = 1,
    CFNotificationSuspensionBehaviorCoalesce = 2,
    CFNotificationSuspensionBehaviorHold = 3,
    CFNotificationSuspensionBehaviorDeliverImmediately = 4
};
typedef enum CFNotificationSuspensionBehavior CFNotificationSuspensionBehavior;
```

**Constants**

CFNotificationSuspensionBehaviorDrop

The server will not queue any notifications of the specified name and object while the receiving application is in the background.

Available in Mac OS X v10.0 and later.

Declared in CFNotificationCenter.h.

CFNotificationSuspensionBehaviorCoalesce

The server will only queue the last notification of the specified name and object; earlier notifications are dropped.

Available in Mac OS X v10.0 and later.

Declared in CFNotificationCenter.h.

CFNotificationSuspensionBehaviorHold

The server will hold all matching notifications until the queue has been filled (queue size determined by the server) at which point the server may flush queued notifications.

Available in Mac OS X v10.0 and later.

Declared in CFNotificationCenter.h.

CFNotificationSuspensionBehaviorDeliverImmediately

The server will deliver notifications of the specified name and object whether or not the application is in the background. When a notification with this suspension behavior is matched, it has the effect of first flushing any queued notifications.

Available in Mac OS X v10.0 and later.

Declared in CFNotificationCenter.h.

**Discussion**

An application selects the suspension behavior for a given notification when it registers an observer for that notification with [CFNotificationCenterAddObserver](#) (page 6).

## Notification Posting Options

Possible options when posting notifications.

```
enum {
    kCFNotificationDeliverImmediately = (1 << 0),
    kCFNotificationPostToAllSessions = (1 << 1)
};
```

### Constants

`kCFNotificationDeliverImmediately`  
Delivers the notification immediately.  
Available in Mac OS X v10.3 and later.  
Declared in `CFNotificationCenter.h`.

`kCFNotificationPostToAllSessions`  
Delivers the notification to all sessions.  
Available in Mac OS X v10.3 and later.  
Declared in `CFNotificationCenter.h`.

### Discussion

Use these constants when calling the [CFNotificationCenterPostNotificationWithOptions](#) (page 10) function.



# Document Revision History

---

This table describes the changes to *CFNotificationCenter Reference*.

Date	Notes
2007-01-22	Noted constraints on object parameter to <code>CFNotificationCenterPostNotification()</code> and on numerous functions when used with a Darwin notification center.
2005-12-06	Minor changes to text to conform to reference consistency guidelines.
2005-11-09	Corrected link to reference book.
2005-08-11	Corrected minor typographical error.
2005-04-29	Updated for Mac OS X v10.4.
	Updated to include new API in Mac OS X version 10.4.
2003-08-01	Modified descriptions per any implementation changes in Mac OS X v10.3.
2003-01-01	First version of this document.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

CFNotificationCenterCallback **callback** [12](#)  
CFNotificationCenterAddObserver **function** [6](#)  
CFNotificationCenterGetDarwinNotifyCenter  
**function** [8](#)  
CFNotificationCenterGetDistributedCenter  
**function** [8](#)  
CFNotificationCenterGetLocalCenter **function** [9](#)  
CFNotificationCenterGetTypeID **function** [9](#)  
CFNotificationCenterPostNotification **function**  
[9](#)  
CFNotificationCenterPostNotificationWithOptions  
**function** [10](#)  
CFNotificationCenterRef **data type** [13](#)  
CFNotificationCenterRemoveEveryObserver  
**function** [11](#)  
CFNotificationCenterRemoveObserver **function** [12](#)  
CFNotificationSuspensionBehaviorCoalesce  
**constant** [14](#)  
CFNotificationSuspensionBehaviorDeliverImmediately  
**constant** [14](#)  
CFNotificationSuspensionBehaviorDrop **constant**  
[14](#)  
CFNotificationSuspensionBehaviorHold **constant**  
[14](#)

## K

---

kCFNotificationDeliverImmediately **constant** [15](#)  
kCFNotificationPostToAllSessions **constant** [15](#)

## N

---

Notification Delivery Suspension Behavior [14](#)  
Notification Posting Options [15](#)