# CFUserNotification Reference

**Core Foundation**

**2006-02-07**

Apple Inc.
© 2003, 2006 Apple Computer, 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.

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


## CFUserNotification Reference 5

Overview 5
Functions 5
- CFUserNotificationCancel 5
- CFUserNotificationCheckBoxChecked 6
- CFUserNotificationCreate 6
- CFUserNotificationCreateRunLoopSource 7
- CFUserNotificationDisplayAlert 8
- CFUserNotificationDisplayNotice 9
- CFUserNotificationGetResponseDictionary 11
- CFUserNotificationGetResponseValue 11
- CFUserNotificationGetTypeID 12
- CFUserNotificationPopUpSelection 12
- CFUserNotificationReceiveResponse 12
- CFUserNotificationSecureTextField 13
- CFUserNotificationUpdate 14

Callbacks 14
- CFUserNotificationCallBack 14

Data Types 15
- CFUserNotificationRef 15

Constants 15
- Alert Levels 15
- Response Codes 16
- Button Flags 17
- Dialog Description Keys 17

## Document Revision History 21

## Index 23




2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

# CFUserNotification Reference

| | |
|---|---|
| **Derived From:** | CFType |
| **Framework:** | CoreFoundation/CoreFoundation.h |
| **Companion guide** | Notification Programming Topics for Cocoa |
| **Declared in** | CFUserNotification.h |

## Overview

A `CFUserNotification` object presents a simple dialog on the screen and optionally receives feedback from the user. The contents of the dialog can include a header, a message, an icon, text fields, a pop-up button, radio buttons or checkboxes, and up to three ordinary buttons. Use `CFUserNotification` in processes that do not otherwise have user interfaces, but may need occasional interaction with the user.

You create a user notification with the `CFUserNotificationCreate` (page 6) function. You pass in a dictionary whose keys describe the items to place into the dialog. (See "Dialog Description Keys" (page 17) for the list of keys.) A set of flags passed to the function determines, among other things, whether secure text fields are used (such as for password fields), whether radio buttons or checkboxes are used, and which of these buttons are checked by default. You can also specify a timeout for the dialog, in which case the dialog cancels itself if the user does not respond in the allotted time period.

A user notification displays its dialog as soon as it is created. If any reply is required, it may be awaited in one of two ways: either synchronously, using `CFUserNotificationReceiveResponse` (page 12), or asynchronously, using a run loop source created with `CFUserNotificationCreateRunLoopSource` (page 7). `CFUserNotificationReceiveResponse` (page 12) has a timeout parameter that determines how long it will block (zero meaning indefinitely) and it may be called as many times as necessary until a response arrives. If a user notification has not yet received a response, it may be updated with new information or it may be cancelled. User notifications may not be reused.

`CFUserNotification` provides two convenience functions, `CFUserNotificationDisplayNotice` (page 9) and `CFUserNotificationDisplayAlert` (page 8), to display very basic dialogs that either require no response from the user or require only a single button to be pressed, respectively.

## Functions

### CFUserNotificationCancel

Cancels a user notification dialog.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

```
SInt32 CFUserNotificationCancel (
   CFUserNotificationRef userNotification
);
```

**Parameters**

*userNotification*

The user notification to cancel.

**Return Value**

`0` if the cancel was successful; a non-`0` value otherwise.

**Discussion**

You must cancel a user notification if you want to remove its dialog from the screen before the user dismisses it. It is not sufficient to just release the object.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CFUserNotification.h`

## CFUserNotificationCheckBoxChecked

Returns a flag used to set or test a checkbox's state.

```
CFOptionFlags CFUserNotificationCheckBoxChecked (
   CFIndex i
);
```

**Parameters**

*idx*

The index of the checkbox to set or test. The index corresponds to the order in which the checkbox titles are listed in the `kCFUserNotificationCheckBoxTitlesKey` (page 19) array of the user notification's description dictionary. *idx* must be in the range `0` to `7`.

**Return Value**

A flag that can be used either to set the state of a checkbox when creating a user notification with `CFUserNotificationCreate` (page 6) or to test a checkbox's state returned in a user notification's response flags, such as from `CFUserNotificationReceiveResponse` (page 12), when the notification dialog is dismissed.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CFUserNotification.h`

## CFUserNotificationCreate

Creates a CFUserNotification object and displays its notification dialog on screen.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

```
CFUserNotificationRef CFUserNotificationCreate (
   CFAllocatorRef allocator,
   CFTimeInterval timeout,
   CFOptionFlags flags,
   SInt32 *error,
   CFDictionaryRef dictionary
);
```

**Parameters**

*allocator*

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

*timeout*

The time to wait before the notification dialog dismisses itself if the user does not respond. If `0`, the notification never times out.

*flags*

A set of flags describing the type of notification to display. These flags specify an alert level for the notification (see Alert Levels (page 15)), determine whether radio buttons or checkboxes are to be used (see Button Flags (page 17)), specify which, if any, of these buttons are checked by default (see `CFUserNotificationCheckBoxChecked` (page 6)), specify whether any of the text fields are to be secure text fields (see `CFUserNotificationSecureTextField` (page 13)), and determine which element of a pop-up menu, if present, should be selected by default (see `CFUserNotificationPopUpSelection` (page 12)). Combine these flags together by performing a bitwise-OR operation with all the individual flags.

*error*

On return contains an integer error code. If `0`, the user notification was successfully created and displayed.

*dictionary*

A description of the elements to display in the notification dialog. The possible keys are listed in "Dialog Description Keys" (page 17). The dictionary must contain a value for the key `kCFUserNotificationAlertHeaderKey` (page 18), but the other keys are optional.

**Return Value**

The new CFUserNotification object. Ownership follows the Create Rule.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CFUserNotification.h`

## CFUserNotificationCreateRunLoopSource

Creates a run loop source for a user notification.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

```
CFRunLoopSourceRef CFUserNotificationCreateRunLoopSource (
   CFAllocatorRef allocator,
   CFUserNotificationRef userNotification,
   CFUserNotificationCallBack callout,
   CFIndex order
);
```

**Parameters**

*allocator*

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

*userNotification*

The user notification to use.

*callout*

The callback function to invoke when the user notification dialog is dismissed.

*order*

A priority index indicating the order in which run loop sources are processed. User notifications currently ignore this parameter. Pass `0` for this value.

**Return Value**

The new CFRunLoopSource object. Ownership follows the Create Rule.

**Discussion**

A run loop source needs to be added to a run loop before it can fire and call its callback function. To add the source to a run loop, use `CFRunLoopAddSource`.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CFUserNotification.h`

## CFUserNotificationDisplayAlert

Displays a user notification dialog and waits for a user response.

```
SInt32 CFUserNotificationDisplayAlert (
   CFTimeInterval timeout,
   CFOptionFlags flags,
   CFURLRef iconURL,
   CFURLRef soundURL,
   CFURLRef localizationURL,
   CFStringRef alertHeader,
   CFStringRef alertMessage,
   CFStringRef defaultButtonTitle,
   CFStringRef alternateButtonTitle,
   CFStringRef otherButtonTitle,
   CFOptionFlags *responseFlags
);
```

**Parameters**

*timeout*

The amount of time to wait for the user to dismiss the notification dialog before the dialog dismisses itself. Pass `0` to have the dialog never time out.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

*flags*

A set of flags describing the type of notification dialog to display. The value is normally just the alert level from Alert Levels (page 15). If you don't want a default button displayed, perform a bitwise-OR operation with the alert level and the constant kCFUserNotificationNoDefaultButtonFlag (page 17).

*iconURL*

A file URL pointing to the icon to display in the dialog. If NULL, a default icon is used based on the notification's alert level specified in *flags*.

*soundURL*

Not used.

*localizationURL*

A file URL pointing to a bundle that contains localized versions of the strings displayed in the dialog. Can be NULL.

*alertHeader*

The title of the notification dialog. Cannot be NULL.

*alertMessage*

The message string to display in the dialog. Can be NULL.

*defaultButtonTitle*

The title of the default button. If NULL, the string OK is used.

*alternateButtonTitle*

The title of an optional alternate button. Can be NULL.

*otherButtonTitle*

The title of an optional third button. Can be NULL.

*responseFlags*

On return, contains flags identifying how the notification was dismissed. See Response Codes (page 16) for details.

**Return Value**
0 if the cancel was successful; a non-0 value otherwise.

**Discussion**
This function blocks the current thread's execution until the dialog is dismissed, either by the user or by timing out.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
CFUserNotification.h

## CFUserNotificationDisplayNotice

Displays a user notification dialog that does not need a user response.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

```
SInt32 CFUserNotificationDisplayNotice (
   CFTimeInterval timeout,
   CFOptionFlags flags,
   CFURLRef iconURL,
   CFURLRef soundURL,
   CFURLRef localizationURL,
   CFStringRef alertHeader,
   CFStringRef alertMessage,
   CFStringRef defaultButtonTitle
);
```

**Parameters**

*timeout*

The amount of time to wait for the user to dismiss the notification dialog before the dialog dismisses itself. Pass `0` to have the dialog never time out.

*flags*

A set of flags describing the type of notification dialog to display. The value is normally just the alert level from Alert Levels (page 15). If you don't want a default button displayed, perform a bitwise-OR operation with the alert level and the constant `kCFUserNotificationNoDefaultButtonFlag` (page 17).

*iconURL*

A file URL pointing to the icon to display in the dialog. If `NULL`, a default icon is used based on the notification's alert level specified in *flags*.

*soundURL*

Not used.

*localizationURL*

A file URL pointing to a bundle that contains localized versions of the strings displayed in the dialog. Can be `NULL`.

*alertHeader*

The title of the notification dialog. Cannot be `NULL`.

*alertMessage*

The message string to display in the dialog. Can be `NULL`.

*defaultButtonTitle*

The title of the default button. If `NULL`, the string `OK` is used.

**Return Value**

`0` if the cancel was successful; a non-`0` value otherwise.

**Discussion**

This function returns immediately. It does not wait for a user response after displaying the dialog.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

X11CallCarbonAndCocoa

**Declared In**

`CFUserNotification.h`


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

## CFUserNotificationGetResponseDictionary

Returns the dictionary containing all the text field values from a dismissed notification dialog.

```
CFDictionaryRef CFUserNotificationGetResponseDictionary (
   CFUserNotificationRef userNotification
);
```

### Parameters

*userNotification*

The user notification to use.

### Return Value

A dictionary holding the values of all the text fields in *userNotification* when it was dismissed. The values are in an array stored with the key kCFUserNotificationTextFieldValuesKey (page 19). Ownership follows the Get Rule.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

CFUserNotification.h

## CFUserNotificationGetResponseValue

Extracts the values of the text fields from a dismissed notification dialog.

```
CFStringRef CFUserNotificationGetResponseValue (
   CFUserNotificationRef userNotification,
   CFStringRef key,
   CFIndex idx
);
```

### Parameters

*userNotification*

The user notification to use.

*key*

The dictionary key identifying the text fields to use. Currently, only kCFUserNotificationTextFieldValuesKey (page 19) is supported.

*idx*

The index of the text field value to return. The index corresponds to the order in which text fields are listed in the kCFUserNotificationTextFieldTitlesKey (page 19) array in the user notification's description dictionary.

### Return Value

The value of the text field identified by *key* and *idx*. Ownership follows the Get Rule.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

CFUserNotification.h


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

## CFUserNotificationGetTypeID

Returns the type identifier for the `CFUserNotification` opaque type.

```
CFTypeID CFUserNotificationGetTypeID (
   void
);
```

**Return Value**
The type identifier for the `CFUserNotification` opaque type.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`CFUserNotification.h`

## CFUserNotificationPopUpSelection

Returns a flag used to set the selected element of a pop-up menu.

```
CFOptionFlags CFUserNotificationPopUpSelection (
   CFIndex n
);
```

**Parameters**

*idx*

The index of the pop-up menu element to select. The index corresponds to the order in which the pop-up menu elements are listed in the `kCFUserNotificationPopUpTitlesKey` (page 19) array of the user notification's description dictionary. *idx* must be in the range `0` to `255`.

**Return Value**
A flag that can be used to set the selected element of a pop-up menu when creating a user notification with `CFUserNotificationCreate` (page 6).

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`CFUserNotification.h`

## CFUserNotificationReceiveResponse

Waits for the user to respond to a notification or for the notification to time out.

```
SInt32 CFUserNotificationReceiveResponse (
   CFUserNotificationRef userNotification,
   CFTimeInterval timeout,
   CFOptionFlags *responseFlags
);
```

**Parameters**

*userNotification*

The user notification to use.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

*timeout*

The amount of time to wait for the user to respond to *userNotification* or for the notification to time out. If neither happens before *timeout* passes, this function returns a non-0 value. If *timeout* is 0, the function blocks until the user notification is dismissed.

*responseFlags*

On return, contains flags identifying how the notification was dismissed, the state of any checkboxes, and the selected element of the pop-up menu. Bits 0-1 of the value hold an identifier for the button pressed by the user (see Response Codes (page 16)). Extract the identifier by performing a bitwise-AND operation with 0x3. Bits 8-15 of *responseFlags* hold the state of up to 8 checkboxes or radio buttons, if present. Extract the flags by performing bitwise-AND operations with the return value of `CFUserNotificationCheckBoxChecked` (page 6). Bits 24-31 hold the index number of the element selected in a pop-up menu, if present. Extract the index by performing a 24-bit right shift: `responseFlags >> 24`.

**Return Value**

0 if the cancel was successful; a non-0 value otherwise.

**Discussion**

Use this function to poll a user notification for a user response. You can call it any number of times on the same user notification.

To avoid polling and blocking your thread's execution, you can create a run loop source for the user notification with `CFUserNotificationCreateRunLoopSource` (page 7). You will then receive a callback when the dialog is dismissed.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CFUserNotification.h

## CFUserNotificationSecureTextField

Returns a flag used to set the secure state of a text field.

```
CFOptionFlags CFUserNotificationSecureTextField (
   CFIndex i
);
```

**Parameters**

*idx*

The index of the text field to make secure. The index corresponds to the order in which the text fields are listed in the `kCFUserNotificationTextFieldTitlesKey` (page 19) array of the user notification's description dictionary. *idx* must be in the range 0 to 7.

**Return Value**

A flag that can be used to set the secure state of a text field when creating a user notification with `CFUserNotificationCreate` (page 6).

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CFUserNotification.h


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

### CFUserNotificationUpdate

Updates a displayed user notification dialog with new user interface information.

```
SInt32 CFUserNotificationUpdate (
   CFUserNotificationRef userNotification,
   CFTimeInterval timeout,
   CFOptionFlags flags,
   CFDictionaryRef dictionary
);
```

**Parameters**

*userNotification*

The user notification to update.

*timeout*

The new timeout value for the dialog.

*flags*

A set of flags describing the type of notification to display. See `CFUserNotificationCreate` (page 6) for details.

*dictionary*

A description of the elements to display in the notification dialog. The possible keys are listed in "Dialog Description Keys" (page 17).

**Return Value**

`0` if the cancel was successful; a non-`0` value otherwise.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CFUserNotification.h`

## Callbacks

### CFUserNotificationCallBack

Callback invoked when an asynchronous user notification dialog is dismissed.

```
typedef void (*CFUserNotificationCallBack) (
    CFUserNotificationRef userNotification,
    CFOptionFlags responseFlags
);
```

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

```
void MyCallBack (
   CFUserNotificationRef userNotification,
   CFOptionFlags responseFlags
);
```


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

**Parameters**

*userNotification*

The user notification that was dismissed.

*responseFlags*

On return, contains flags identifying how the notification was dismissed, the state of any checkboxes, and the selected item of the pop-up menu. See `CFUserNotificationReceiveResponse` (page 12) for details.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`CFUserNotification.h`

# Data Types

### CFUserNotificationRef

A reference to a user notification object.

```
typedef struct __CFUserNotification *CFUserNotificationRef;
```

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`CFUserNotification.h`

# Constants

## Alert Levels

Flags identifying the seriousness of a user notification.

```
enum {
    kCFUserNotificationStopAlertLevel = 0,
    kCFUserNotificationNoteAlertLevel = 1,
    kCFUserNotificationCautionAlertLevel = 2,
    kCFUserNotificationPlainAlertLevel= 3
};
```

**Constants**

`kCFUserNotificationStopAlertLevel`

The notification is very serious.

A stop icon is displayed by default.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

`kCFUserNotificationNoteAlertLevel`
The notification is not very serious.

A note icon is displayed by default.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationCautionAlertLevel`
The notification is somewhat serious.

A caution icon is displayed by default.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationPlainAlertLevel`
The notification is not serious.

An information icon is displayed by default.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

**Discussion**
If you specify an icon to display in the dialog, this icon overrides the default icon used for each alert level.

## Response Codes

Response codes identifying the button that was pressed to dismiss a notification dialog.

```
enum {
    kCFUserNotificationDefaultResponse = 0,
    kCFUserNotificationAlternateResponse = 1,
    kCFUserNotificationOtherResponse = 2,
    kCFUserNotificationCancelResponse = 3
};
```

**Constants**
`kCFUserNotificationDefaultResponse`
The default button was pressed.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationAlternateResponse`
The alternate button was pressed.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationOtherResponse`
The third button was pressed.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

`kCFUserNotificationCancelResponse`
No button was pressed and the notification timed out.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

**Discussion**
To extract this value from the response flags of a dismissed notification (such as returned by `CFUserNotificationReceiveResponse` (page 12)), you must perform a bitwise-AND operation between the returned response flags and `0x3` before comparing the value to these constants.

## Button Flags

Flags that alter the display of buttons in a user notification dialog.

```
enum {
    kCFUserNotificationNoDefaultButtonFlag = (1 << 5),
    kCFUserNotificationUseRadioButtonsFlag = (1 << 6)
};
```

**Constants**
`kCFUserNotificationNoDefaultButtonFlag`
Displays the dialog without the default, alternate, or other buttons.

The dialog remains on screen until it times out or you cancel it with `CFUserNotificationCancel` (page 5). If you provide a title for the default button in the user notification's description dictionary, this flag is ignored and buttons show up normally.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationUseRadioButtonsFlag`
Creates a group of radio buttons instead of checkboxes for the elements in the `kCFUserNotificationCheckBoxTitlesKey` (page 19) array in the user notification's description dictionary.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

**Discussion**
You specify these flags when you create the user notification with `CFUserNotificationCreate` (page 6).

## Dialog Description Keys

Keys used in a user notification's description dictionary, which describes the contents of the notification dialog to display.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

```
const CFStringRef kCFUserNotificationIconURLKey;
const CFStringRef kCFUserNotificationSoundURLKey;
const CFStringRef kCFUserNotificationLocalizationURLKey;
const CFStringRef kCFUserNotificationAlertHeaderKey;
const CFStringRef kCFUserNotificationAlertMessageKey;
const CFStringRef kCFUserNotificationDefaultButtonTitleKey;
const CFStringRef kCFUserNotificationAlternateButtonTitleKey;
const CFStringRef kCFUserNotificationOtherButtonTitleKey;
const CFStringRef kCFUserNotificationProgressIndicatorValueKey;
const CFStringRef kCFUserNotificationPopUpTitlesKey;
const CFStringRef kCFUserNotificationTextFieldTitlesKey;
const CFStringRef kCFUserNotificationCheckBoxTitlesKey;
const CFStringRef kCFUserNotificationTextFieldValuesKey;
const CFStringRef kCFUserNotificationPopUpSelectionKey
```

**Constants**

`kCFUserNotificationIconURLKey`

A file URL pointing to the icon to display in the dialog.

If absent, a default icon based on the alert level is used.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationSoundURLKey`

Not used.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationLocalizationURLKey`

A file URL pointing to a bundle that contains localized versions of the strings displayed in the dialog.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationAlertHeaderKey`

The title of the notification dialog.

This key is required.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationAlertMessageKey`

The message string to display in the dialog.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationDefaultButtonTitleKey`

The title of the default button.

If absent and the dialog is not being created with the `kCFUserNotificationNoDefaultButtonFlag` (page 17) flag, a default button title of `OK` is used.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

`kCFUserNotificationAlternateButtonTitleKey`
The title of an optional alternate button.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationOtherButtonTitleKey`
The title of an optional third button.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationProgressIndicatorValueKey`
Not used.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationPopUpTitlesKey`
The list of strings to display in a pop-up menu.

The array cannot have more than 256 elements.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationTextFieldTitlesKey`
The list of titles for all the text fields to display.

If only one text field is to be displayed, you can pass its title string directly without putting it into an array first.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationCheckBoxTitlesKey`
The list of titles for all the checkboxes or radio buttons to display.

The array cannot have more than 8 elements. If only one checkbox is to be displayed, you can pass its title string directly without putting it into an array first.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationTextFieldValuesKey`
The list of values to put into the text fields. If only one text field is to be displayed, you can pass its value string directly without putting it into an array first.

Available in Mac OS X v10.0 and later.

Declared in `CFUserNotification.h`.

`kCFUserNotificationPopUpSelectionKey`
The item that was selected from a pop-up menu.

Available in Mac OS X v10.3 and later.

Declared in `CFUserNotification.h`.

**Discussion**
When creating the user notification with `CFUserNotificationCreate` (page 6), the description dictionary must have a value for `kCFUserNotificationAlertHeaderKey` (page 18). All other keys are optional.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

The button title keys must be given in right-to-left order—you therefore must use the `kCFUserNotificationDefaultButtonTitleKey` constant for the rightmost button even if it is conceptually not a "default" button (for example, if you want a single "Cancel" button that should not have color, should not pulse, and should not have return for a key equivalent). If, however, you set the `kCFUserNotificationNoDefaultButtonFlag`, the rightmost button does not behave as a default button (although it will still be the "default" button in the sense of using `kCFUserNotificationDefaultButtonTitleKey` and `kCFUserNotificationDefaultResponse`). The following code fragment shows how you can create a notification that contains a single "Cancel" button that does not behave as a default button.

```
const void* keys[] = {kCFUserNotificationAlertHeaderKey,
                      kCFUserNotificationProgressIndicatorValueKey,
                      kCFUserNotificationDefaultButtonTitleKey};
const void* values[] = {CFSTR("Progress"),
                        kCFBooleanTrue,
                        CFSTR("Cancel")};
CFDictionaryRef parameters = CFDictionaryCreate(0, keys, values,
        sizeof(keys)/sizeof(*keys), &kCFTypeDictionaryKeyCallBacks,
        &kCFTypeDictionaryValueCallBacks);
SInt32 err = 0;
CFUserNotificationCreate(kCFAllocatorDefault, 0,
        kCFUserNotificationPlainAlertLevel |
kCFUserNotificationNoDefaultButtonFlag,
        &err, parameters);
```

If you set the `kCFUserNotificationNoDefaultButtonFlag` flag and do not specify a value for `kCFUserNotificationDefaultButtonTitleKey`, the notification will have no buttons.


2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

# Document Revision History

This table describes the changes to *CFUserNotification Reference*.

| Date | Notes |
| --- | --- |
| 2006-02-07 | Clarified use of kCFUserNotificationDefaultButtonTitleKey for showing alternate buttons. |
| 2005-11-09 | Corrected link to reference document. |
| 2005-08-11 | Cosmetic changes to conform to documentation guidelines. |
| 2003-08-01 | Added descriptions for new Mac OS X v10.3 API. |
| 2003-01-01 | First version of this document. |



2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

Document Revision History



2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.

# Index


## A

Alert Levels 15

## B

Button Flags 17

## C

CFUserNotificationCallBack callback 14
CFUserNotificationCancel function 5
CFUserNotificationCheckBoxChecked function 6
CFUserNotificationCreate function 6
CFUserNotificationCreateRunLoopSource function 7
CFUserNotificationDisplayAlert function 8
CFUserNotificationDisplayNotice function 9
CFUserNotificationGetResponseDictionary function 11
CFUserNotificationGetResponseValue function 11
CFUserNotificationGetTypeID function 12
CFUserNotificationPopUpSelection function 12
CFUserNotificationReceiveResponse function 12
CFUserNotificationRef data type 15
CFUserNotificationSecureTextField function 13
CFUserNotificationUpdate function 14

## D

Dialog Description Keys 17

## K

kCFUserNotificationAlertHeaderKey constant 18
kCFUserNotificationAlertMessageKey constant 18
kCFUserNotificationAlternateButtonTitleKey constant 19
kCFUserNotificationAlternateResponse constant 16
kCFUserNotificationCancelResponse constant 17
kCFUserNotificationCautionAlertLevel constant 16
kCFUserNotificationCheckBoxTitlesKey constant 19
kCFUserNotificationDefaultButtonTitleKey constant 18
kCFUserNotificationDefaultResponse constant 16
kCFUserNotificationIconURLKey constant 18
kCFUserNotificationLocalizationURLKey constant 18
kCFUserNotificationNoDefaultButtonFlag constant 17
kCFUserNotificationNoteAlertLevel constant 16
kCFUserNotificationOtherButtonTitleKey constant 19
kCFUserNotificationOtherResponse constant 16
kCFUserNotificationPlainAlertLevel constant 16
kCFUserNotificationPopUpSelectionKey constant 19
kCFUserNotificationPopUpTitlesKey constant 19
kCFUserNotificationProgressIndicatorValueKey constant 19
kCFUserNotificationSoundURLKey constant 18
kCFUserNotificationStopAlertLevel constant 15
kCFUserNotificationTextFieldTitlesKey constant 19
kCFUserNotificationTextFieldValuesKey constant 19
kCFUserNotificationUseRadioButtonsFlag constant 17

## R

Response Codes 16




2006-02-07 | © 2003, 2006 Apple Computer, Inc. All Rights Reserved.