
CFError Reference

[Core Foundation](#) > [Events & Other Input](#)



2006-07-12



Apple Inc.
© 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, and Mac 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

CFError Reference 5

Overview	5
Functions by Task	5
Creating a CFError	5
Getting Information About an Error	6
Getting the CFError Type ID	6
Functions	6
CFErrorCopyDescription	6
CFErrorCopyFailureReason	7
CFErrorCopyRecoverySuggestion	8
CFErrorCopyUserInfo	8
CFErrorCreate	9
CFErrorCreateWithUserInfoKeysAndValues	9
CFErrorGetCode	10
CFErrorGetDomain	11
CFErrorGetTypeID	11
Data Types	12
CFErrorRef	12
Constants	12
Error domains	12
Keys for the user info dictionary	13

Document Revision History 15

Index 17

CFError Reference

Derived From:	CType
Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Error Handling Programming Guide For Cocoa
Declared in	CFError.h

Overview

A CFError object encapsulates rich and extensible error information than is possible using only an error code or error string. The core attributes of a CFError object are an error domain (represented by a string), a domain-specific error code and a user info dictionary containing application specific information. Errors are required to have a domain and an error code within that domain. The optional "userInfo" dictionary may provide additional information that might be useful for the interpretation and reporting of the error. This dictionary can even contain an "underlying" error, which is wrapped as an error bubbles up through various layers.

Several well-known domains are defined corresponding to Mach, POSIX, and OSStatus errors. In addition, CFError allows you to attach an arbitrary user info dictionary to an error object, and provides the means to return a human-readable description for the error.

In general, a method should signal an error condition by—for example—returning `false` or `NULL` rather than by the simple presence of an error object. The method can then optionally return a CFError object by reference, in order to further describe the error.

CFError is toll-free bridged to NSError in the Foundation framework—for more details on toll-free bridging, see *Interchangeable Data Types*. NSError has some additional guidelines which makes it easy to automatically report errors to users and even try to recover from them. See *Error Handling Programming Guide For Cocoa* for more information on NSError programming guidelines.

Functions by Task

Creating a CFError

[CFErrorCreate](#) (page 9)

Creates a new CFError object.

[CFErrorCreateWithUserInfoKeysAndValues](#) (page 9)

Creates a new CFError object using given keys and values to create the user info dictionary.

Getting Information About an Error

[CFErrorGetDomain](#) (page 11)

Returns the error domain for a given CFError.

[CFErrorGetCode](#) (page 10)

Returns the error code for a given CFError.

[CFErrorCopyUserInfo](#) (page 8)

Returns the user info dictionary for a given CFError.

[CFErrorCopyDescription](#) (page 6)

Returns a human-presentable description for a given error.

[CFErrorCopyFailureReason](#) (page 7)

Returns a human-presentable failure reason for a given error.

[CFErrorCopyRecoverySuggestion](#) (page 8)

Returns a human presentable recovery suggestion for a given error.

Getting the CFError Type ID

[CFErrorGetTypeID](#) (page 11)

Returns the type identifier for the CFError opaque type.

Functions

CFErrorCopyDescription

Returns a human-presentable description for a given error.

```
CFStringRef CFErrorCopyDescription (
    CFErrorRef err
);
```

Parameters

err

The CFError to examine. If this is not a valid CFError, the behavior is undefined.

Return Value

A localized, human-presentable description of *err*. This function never returns `NULL`. Ownership follows the Create Rule.

Discussion

This is a complete sentence or two which says what failed and why it failed. The structure of the description depends on the details provided in the user info dictionary. The rules for computing the return value are as follows:

1. If the value in the user info dictionary for `kCFErrorLocalizedDescriptionKey` (page 13) is not `NULL`, returns that value as-is.
2. If the value in the user info dictionary for `kCFErrorLocalizedFailureReasonKey` (page 13) is not `NULL`, generate an error from that.

The description is something like: "Operation could not be completed." + `kCFErrorLocalizedFailureReasonKey`

3. Generate as good a user-presentable string as possible from `kCFErrorDescriptionKey` (page 13), the domain, and code.

The description is something like like: "Operation could not be completed. Error domain/code occurred." or "Operation could not be completed." + `kCFErrorDescriptionKey` + " (Error domain/code)"

Toll-free bridged instances of `NSError` might provide additional behaviors for manufacturing a description string.

You should not depend on the exact contents or format of the returned string, as it might change in different releases of the operating system.

When you create a `CFError`, you should try to make sure the return value is human-presentable and localized by providing a value for `kCFErrorLocalizedDescriptionKey` (page 13) in the user info dictionary.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CFError.h`

CFErrorCopyFailureReason

Returns a human-presentable failure reason for a given error.

```
CFStringRef CFErrorCopyFailureReason (
    CFErrorRef err
);
```

Parameters

err

The `CFError` to examine. If this is not a valid `CFError`, the behavior is undefined.

Return Value

A localized, human-presentable failure reason for *err*, or `NULL` if no user-presentable string is available. Ownership follows the Create Rule.

Discussion

The failure reason is a complete sentence which describes why the operation failed. In many cases this will be just the "because" part of the description (but as a complete sentence, which makes localization easier). For example, an error description "Could not save file 'Letter' in folder 'Documents' because the volume 'MyDisk' doesn't have enough space." might have a corresponding failure reason, "The volume 'MyDisk' doesn't have enough space."

By default, this function looks for a value for the `kCFErrorLocalizedFailureReasonKey` (page 13) key in the user info dictionary. Toll-free bridged instances of `NSError` might provide additional behaviors for manufacturing this value.

When you create a `CFError`, you should try to make sure the return value is human-presentable and localized by providing a value for `kCFErrorLocalizedFailureReasonKey` (page 13) in the user info dictionary.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorCopyRecoverySuggestion

Returns a human presentable recovery suggestion for a given error.

```
CFStringRef CFErrorCopyRecoverySuggestion (
    CFErrorRef err
);
```

Parameters

err

The CFError to examine. If this is not a valid CFError, the behavior is undefined.

Return Value

A localized, human-presentable recovery suggestion for *err*, or NULL if no user-presentable string is available. Ownership follows the Create Rule.

Discussion

This is the string that can be displayed as the “informative” (or “secondary”) message on an alert panel. For example, an error description “Could not save file ‘Letter’ in folder ‘Documents’ because the volume ‘MyDisk’ doesn’t have enough space.” might have a corresponding recovery suggestion, “Remove some files from the volume and try again.”

By default, this function looks for a value for the [kCFErrorLocalizedRecoverySuggestionKey](#) (page 13) key in the user info dictionary. Toll-free bridged instances of NSError might provide additional behaviors for manufacturing this value.

When you create a CFError, you should try to make sure the return value is human-presentable and localized by providing a value for [kCFErrorLocalizedRecoverySuggestionKey](#) (page 13) in the user info dictionary.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorCopyUserInfo

Returns the user info dictionary for a given CFError.

```
CFDictionaryRef CFErrorCopyUserInfo (
    CFErrorRef err
);
```

Parameters

err

The error to examine. If this is not a valid CFError, the behavior is undefined.

Return Value

A dictionary containing the same keys and values as in the userInfo dictionary *err* was created with. Returns an empty dictionary if NULL was supplied to the create function. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorCreate

Creates a new CFError object.

```
CFErrorRef CFErrorCreate (
    CFAllocatorRef allocator,
    CFStringRef domain,
    CFIndex code,
    CFDictionaryRef userInfo
);
```

Parameters

allocator

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

domain

A CFString that identifies the error domain. If this reference is NULL or is otherwise not a valid CFString, the behavior is undefined.

code

A CFIndex that identifies the error code. The code is interpreted within the context of the error domain.

userInfo

A CFDictionary created with `kCFCopyStringDictionaryKeyCallBacks` and `kCFTypeDictionaryValueCallBacks`. The dictionary is copied with `CFDictionaryCreateCopy`. If you do not want the userInfo dictionary, you can pass NULL, in which case an empty dictionary will be assigned.

Return Value

A new CFError object. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorCreateWithUserInfoKeysAndValues

Creates a new CFError object using given keys and values to create the user info dictionary.

```
CFErrorRef CFErrorCreateWithUserInfoKeysAndValues (
    CFAllocatorRef allocator,
    CFStringRef domain,
    CFIndex code,
    const void *const *userInfoKeys,
    const void *const *userInfoValues,
    CFIndex numUserInfoValues
);
```

Parameters*allocator*

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

domain

A `CFStringRef` that identifies the error domain. If this reference is `NULL` or is otherwise not a valid `CFString`, the behavior is undefined.

code

A `CFIndex` that identifies the error code. The code is interpreted within the context of the error domain.

userInfoKeys

An array of *numUserInfoValues* `CFStrings` used as keys in creating the userInfo dictionary. The value of this parameter can be `NULL` if *numUserInfoValues* is 0.

userInfoValues

An array of *numUserInfoValues* `CF types` used as values in creating the userInfo dictionary. The value of this parameter can be `NULL` if *numUserInfoValues* is 0.

numUserInfoValues

The number of keys and values in the *userInfoKeys* and *userInfoValues* arrays.

Return Value

A new `CFError` object. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CFError.h`

CFErrorGetCode

Returns the error code for a given `CFError`.

```
CFIndex CFErrorGetCode (
    CFErrorRef err
);
```

Parameters*err*

The error to examine. If this is not a valid `CFError`, the behavior is undefined.

Return Value

The error code of *err*.

Discussion

Note that this function returns the error code for the specified CFError, not an error return for the current call.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorGetDomain

Returns the error domain for a given CFError.

```
CFStringRef CFErrorGetDomain (  
    CFErrorRef err  
);
```

Parameters

err

The error to examine. If this is not a valid CFError, the behavior is undefined.

Return Value

The error domain for *err*. Ownership follows the Get Rule.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

CFErrorGetTypeID

Returns the type identifier for the CFError opaque type.

```
CTypeID CFErrorGetTypeID (  
    void  
);
```

Return Value

The type identifier for the CFError opaque type.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

Data Types

CFErrorRef

A reference to a CFError object.

```
typedef struct __CFError * CFErrorRef;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

Constants

Error domains

These constants define domains for CFError objects.

```
const CFStringRef kCFErrorDomainPOSIX;
const CFStringRef kCFErrorDomainOSStatus;
const CFStringRef kCFErrorDomainMach;
const CFStringRef kCFErrorDomainCocoa;
```

Constants

`kCFErrorDomainPOSIX`

A constant that specified the POSIX domain.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

`kCFErrorDomainOSStatus`

A constant that specified the OS domain.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

`kCFErrorDomainMach`

A constant that specified the Mach domain.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

`kCFErrorDomainCocoa`

A constant that specified the Cocoa domain.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

Discussion

The value of "code" will correspond to preexisting values in these domains.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CFError.h

Keys for the user info dictionary

Keys in the userInfo dictionary.

```
const CFStringRef kCFErrorLocalizedDescriptionKey;
const CFStringRef kCFErrorLocalizedFailureReasonKey;
const CFStringRef kCFErrorLocalizedRecoverySuggestionKey;
const CFStringRef kCFErrorDescriptionKey;
const CFStringRef kCFErrorUnderlyingErrorKey;
```

Constants

kCFErrorLocalizedDescriptionKey

Key to identify the end user-presentable description in the userInfo dictionary.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

kCFErrorLocalizedFailureReasonKey

Key to identify the end user-presentable failure reason in the userInfo dictionary.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

kCFErrorLocalizedRecoverySuggestionKey

Key to identify the end user-presentable recovery suggestion in the userInfo dictionary.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

kCFErrorDescriptionKey

Key to identify the description in the userInfo dictionary.

When you create a CFError, you can provide a value for this key if you do not have localizable error strings. The description should be a complete sentence if possible, and should not contain the domain name or error code.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

kCFErrorUnderlyingErrorKey

Key to identify the underlying error in the userInfo dictionary.

Available in Mac OS X v10.5 and later.

Declared in CFError.h.

Discussion

When you create a user info dictionary, at a minimum you should provide values for one of kCFErrorLocalizedDescriptionKey and kCFErrorLocalizedFailureReasonKey; ideally you should provide values for kCFErrorLocalizedDescriptionKey, kCFErrorLocalizedFailureReasonKey, and kCFErrorLocalizedRecoverySuggestionKey.

Availability

Available in Mac OS X v10.5 and later.

Declared In
NSError.h

Document Revision History

This table describes the changes to *CFError Reference*.

Date	Notes
2006-07-12	New document that describes the opaque type used to represent error information in Core Foundation.

REVISION HISTORY

Document Revision History

Index

C

CFErrorCopyDescription **function** [6](#)
CFErrorCopyFailureReason **function** [7](#)
CFErrorCopyRecoverySuggestion **function** [8](#)
CFErrorCopyUserInfo **function** [8](#)
CFErrorCreate **function** [9](#)
CFErrorCreateWithUserInfoKeysAndValues **function**
[9](#)
CFErrorGetCode **function** [10](#)
CFErrorGetDomain **function** [11](#)
CFErrorGetTypeID **function** [11](#)
CFErrorRef **data type** [12](#)

E

Error domains [12](#)

K

kCFErrorDescriptionKey **constant** [13](#)
kCFErrorDomainCocoa **constant** [12](#)
kCFErrorDomainMach **constant** [12](#)
kCFErrorDomainOSStatus **constant** [12](#)
kCFErrorDomainPOSIX **constant** [12](#)
kCFErrorLocalizedDescriptionKey **constant** [13](#)
kCFErrorLocalizedFailureReasonKey **constant** [13](#)
kCFErrorLocalizedRecoverySuggestionKey
constant [13](#)
kCFErrorUnderlyingErrorKey **constant** [13](#)
Keys for the user info dictionary [13](#)