
Core Foundation URL Access Utilities Reference

Core Foundation



2007-01-19



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, Leopard, 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

Core Foundation URL Access Utilities Reference 5

Overview	5
Functions	5
CFURLCreateDataAndPropertiesFromResource	5
CFURLCreatePropertyFromResource	6
CFURLDestroyResource	7
CFURLWriteDataAndPropertiesToResource	8
Constants	9
Error Codes	9
File URL Properties	10
HTTP URL Properties	11

Document Revision History 13

Index 15

Core Foundation URL Access Utilities Reference

Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFURLAccess.h

Overview

Core Foundation URL Access Utilities give you convenient system-independent methods of creating, reading, updating, or deleting a URL resource.

Given a CFURL object that holds either a file or http URL, you can read the resource's data with the [CFURLCreateDataAndPropertiesFromResource](#) (page 5) function. You can write data to the URL resource, possibly creating a new file, with the [CFURLWriteDataAndPropertiesToResource](#) (page 8) function. Finally, you can destroy, or delete, the resource pointed to by the URL with the [CFURLDestroyResource](#) (page 7) function.

Functions

CFURLCreateDataAndPropertiesFromResource

Loads the data and properties referred to by a given URL.

```
Boolean CFURLCreateDataAndPropertiesFromResource (
    CFAllocatorRef alloc,
    CFURLRef url,
    CFDataRef *resourceData,
    CFDictionaryRef *properties,
    CFArrayRef desiredProperties,
    SInt32 *errorCode
);
```

Parameters

allocator

The allocator to use to allocate memory for the new CFData and CFDictionary objects returned in *resourceData* and *properties*. Pass NULL or kCFAllocatorDefault to use the current default allocator.

url

The URL referring to the data and/or properties you wish to load.

resourceData

On return, contains a CFData object containing the data referred to by *url*. Ownership follows the Create Rule.

properties

On return, a pointer to a `CFDictionary` object containing the resource properties referred to by *url*. Ownership follows the Create Rule.

desiredProperties

A list of the properties you wish to obtain and return in *properties*. See “File URL Properties” (page 10) and “HTTP URL Properties” (page 11) for the list of available properties.

errorCode

0 if successful, otherwise an error code indicating the nature of the problem. See “Error Codes” (page 9) for a list of possible error codes.

Return Value

true if successful, false otherwise.

Discussion

If you are interested in loading only the resource data or the resource's properties, pass `NULL` for the one you don't want. If *properties* is non-`NULL` and *desiredProperties* is `NULL` then all properties are fetched. Note that as much work as possible is done even if `false` is returned. For instance, if one property is not available, the others are fetched anyway. This function is intended for convenience, not performance.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

GLCarbon1ContextPbuffer

HID Config Save

HID Utilities Source

MoreIsBetter

QISA

Declared In

`CFURLAccess.h`

CFURLCreatePropertyFromResource

Returns a given property specified by a given URL and property string.

```

CTypeRef CFURLCreatePropertyFromResource (
    CFAllocatorRef alloc,
    CFURLRef url,
    CFStringRef property,
    SInt32 *errorCode
);

```

Parameters

allocator

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

url

The `CFURL` object referring to the resource whose properties are loaded.

property

The name of the property you wish to load. Pass one of the provided string constants indicating the property. See [“File URL Properties”](#) (page 10) and [“HTTP URL Properties”](#) (page 11) for the list of available properties.

errorCode

On return, 0 if successful, otherwise an error code indicating the nature of the problem. See [“Error Codes”](#) (page 9) for a list of possible error codes.

Return Value

If successful, the requested property as a `CFTYPE` object, `NULL` otherwise. Ownership follows the Create Rule.

Discussion

This is a convenience function for retrieving individual property values which calls through to [`CFURLCreateDataAndPropertiesFromResource`](#) (page 5).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFURLAccess.h`

CFURLDestroyResource

Destroys a resource indicated by a given URL.

```
Boolean CFURLDestroyResource (
    CFURLRef url,
    SInt32 *errorCode
);
```

Parameters

url

The `CFURL` object of the resource to destroy.

errorCode

On return, 0 if successful, otherwise an error code indicating the nature of the problem. See [“Error Codes”](#) (page 9) for a list of possible error codes.

Return Value

`true` if successful, `false` otherwise.

Discussion

If *url* uses an `http` scheme, an `http DELETE` request is sent to the resource. If *url* uses a file scheme, then:

- if the reference is a file, the file is deleted;
- if the reference is a directory and the directory is empty, the directory is deleted;
- if the reference is a directory and the directory is not empty, the function returns `false` and *errorCode* contains `kCFURLUnknownError`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFURLAccess.h`

CFURLWriteDataAndPropertiesToResource

Writes the given data and properties to a given URL.

```

Boolean CFURLWriteDataAndPropertiesToResource (
    CFURLRef url,
    CFDataRef dataToWrite,
    CFDictionaryRef propertiesToWrite,
    SInt32 *errorCode
);

```

Parameters

url

The resource to write.

dataToWrite

The data to write. Pass NULL to write only properties.

propertiesToWrite

The properties to write. Pass NULL to write only data. See “File URL Properties” (page 10) and “HTTP URL Properties” (page 11) for the list of available properties.

errorCode

Upon return, 0 if successful, otherwise contains an error code indicating the nature of the problem. See “Error Codes” (page 9) for a list of possible error codes.

Return Value

true if successful, false otherwise.

Discussion

Properties not present in *propertiesToWrite* are left unchanged, hence if *propertiesToWrite* is NULL or empty, the URL's properties are not changed at all.

If *url* uses a file scheme and it references a file, the contents of *dataToWrite* are written to the referenced file, overwriting any preexisting data, and the file's properties are modified according to *propertiesToWrite*. If the file does not exist, but all intermediate directories along the path do already exist, the file is created (otherwise it is not).

If *url* uses a file scheme and it references a directory (the last path character is "/"), the contents of *dataToWrite* are ignored, but if the parameter value is not NULL—and all intermediate directories along the path do already exist—a new directory is created (otherwise it is not).

If *url* uses an http scheme, an http PUT request is sent to the resource with *propertiesToWrite* as the header fields and *dataToWrite* as the data.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CFPrefTopScores

ImageBrowserView

MoreIsBetter

SeeMyFriends

Declared In

CFURLAccess.h

Constants

Error Codes

CFURL error codes.

```
enum CFURLError {
    kCFURLUnknownError = -10,
    kCFURLUnknownSchemeError = -11,
    kCFURLResourceNotFoundError = -12,
    kCFURLResourceAccessViolationError = -13,
    kCFURLRemoteHostUnavailableError = -14,
    kCFURLImproperArgumentsError = -15,
    kCFURLUnknownPropertyKeyError = -16,
    kCFURLPropertyKeyUnavailableError = -17,
    kCFURLTimeoutError = -18
};
typedef enum CFURLError CFURLError;
```

Constants

`kCFURLUnknownError`

Indicates an unknown error.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLUnknownSchemeError`

Indicates that the scheme is not recognized.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLResourceNotFoundError`

Indicates a resource was not found.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLResourceAccessViolationError`

Indicates an error in accessing a resource.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLRemoteHostUnavailableError`

Indicates a remote host is unavailable.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLImproperArgumentsError`

Indicates one or more arguments are improper.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLUnknownPropertyKeyError`

Indicates a property key is unknown.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLPropertyKeyUnavailableError`

Indicates a property key was unavailable.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLTimeoutError`

Indicates a timeout.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

Availability

Available in Mac OS X version 10.0 and later.

Declared In

`CFURLAccess.h`

File URL Properties

Properties for file URL resources.

```
const CFStringRef kCFURLFileExists;
const CFStringRef kCFURLFileDirectoryContents;
const CFStringRef kCFURLFileLength;
const CFStringRef kCFURLFileLastModificationTime;
const CFStringRef kCFURLFilePOSIXMode;
const CFStringRef kCFURLFileOwnerID;
```

Constants

`kCFURLFileExists`

A `CFBoolean` object indicating whether the file referred to by a URL exists.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLFileDirectoryContents`

A `CFArray` object holding `CFURL` objects for the contents of a directory referred to by a URL.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLFileLength`

A `CFNumber` object holding the file's length in bytes.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLFileLastModificationTime`

A `CFDate` object holding the file's modification time.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLFilePOSIXMode`

A `CFNumber` holding the file's POSIX mode as given in `/usr/include/sys/stat.h`.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLFileOwnerID`

A `CFNumber` holding the file owner's UID.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

Availability

Available in Mac OS X version 10.0 and later.

Declared In

`CFURLAccess.h`

HTTP URL Properties

Properties for HTTP URL resources.

```
const CFStringRef kCFURLHTTPStatusCode;  
const CFStringRef kCFURLHTTPStatusLine;
```

Constants

`kCFURLHTTPStatusCode`

A `CFNumber` object holding the status code of an HTTP request.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

`kCFURLHTTPStatusLine`

A `CFString` object holding the status line of an HTTP request.

Available in Mac OS X v10.0 and later.

Declared in `CFURLAccess.h`.

Discussion

In addition to the above properties, each field within an HTTP request or response header is itself a property. You can specify a header field by using the field name as the property name.

Availability

Available in Mac OS X version 10.0 and later.

Declared In

`CFURLAccess.h`

Document Revision History

This table describes the changes to *Core Foundation URL Access Utilities Reference*.

Date	Notes
2007-01-19	Clarified behavior of CFURLDestroyResource.
	If any additional changes are made for Leopard, add revision history item for 2.4.
2006-01-10	Clarified the description of the CFURLWriteDataAndPropertiesToResource function.
2005-11-09	Removed reference to retired document.
2005-08-11	Cosmetic changes to conform to documentation guidelines.
2003-01-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

C

CFURLCreateDataAndPropertiesFromResource **function** [5](#)
CFURLCreatePropertyFromResource **function** [6](#)
CFURLDestroyResource **function** [7](#)
CFURLWriteDataAndPropertiesToResource **function** [8](#)

kCFURLResourceNotFoundError **constant** [9](#)
kCFURLTimeoutError **constant** [10](#)
kCFURLUnknownError **constant** [9](#)
kCFURLUnknownPropertyKeyError **constant** [10](#)
kCFURLUnknownSchemeError **constant** [9](#)

E

Error Codes [9](#)

F

File URL Properties [10](#)

H

HTTP URL Properties [11](#)

K

kCFURLFileDirectoryContents **constant** [10](#)
kCFURLFileExists **constant** [10](#)
kCFURLFileLastModificationTime **constant** [10](#)
kCFURLFileLength **constant** [10](#)
kCFURLFileOwnerID **constant** [11](#)
kCFURLFilePOSIXMode **constant** [11](#)
kCFURLHTTPStatusCode **constant** [11](#)
kCFURLHTTPStatusLine **constant** [11](#)
kCFURLImproperArgumentsError **constant** [9](#)
kCFURLPropertyKeyUnavailableError **constant** [10](#)
kCFURLRemoteHostUnavailableError **constant** [9](#)
kCFURLResourceAccessViolationError **constant** [9](#)