
CFURL Reference

Networking, Internet, & Web: Protocol Streams



2009-08-07



Apple Inc.
© 2003, 2009 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, Carbon, Cocoa, iTunes, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

Aperture and Finder are trademarks 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

CFURL Reference 7

Overview	7
Functions by Task	7
Creating a CFURL	7
Accessing the Parts of a URL	8
Converting URLs to Other Representations	9
Getting URL Properties	10
Getting and Setting File System Resource Properties	10
Working with Bookmark Data	10
Functions	11
CFURLCanBeDecomposed	11
CFURLClearResourcePropertyCache	11
CFURLClearResourcePropertyCacheForKey	12
CFURLCopyAbsoluteURL	12
CFURLCopyFilePath	12
CFURLCopyFragment	13
CFURLCopyHostName	14
CFURLCopyLastPathComponent	14
CFURLCopyNetLocation	15
CFURLCopyParameterString	16
CFURLCopyPassword	16
CFURLCopyPath	17
CFURLCopyPathExtension	18
CFURLCopyQueryString	18
CFURLCopyResourcePropertiesForKeys	19
CFURLCopyResourcePropertyForKey	20
CFURLCopyResourceSpecifier	20
CFURLCopyScheme	21
CFURLCopyStrictPath	21
CFURLCopyUserName	22
CFURLCreateAbsoluteURLWithBytes	23
CFURLCreateBookmarkData	23
CFURLCreateBookmarkDataFromAliasRecord	24
CFURLCreateBookmarkDataFromFile	25
CFURLCreateByResolvingBookmarkData	25
CFURLCreateCopyAppendingPathComponent	26
CFURLCreateCopyAppendingPathExtension	27
CFURLCreateCopyDeletingLastPathComponent	28
CFURLCreateCopyDeletingPathExtension	28
CFURLCreateData	29
CFURLCreateFilePathURL	30

- CFURLCreateFileReferenceURL 30
- CFURLCreateFromFileSystemRepresentation 31
- CFURLCreateFromFileSystemRepresentationRelativeToBase 31
- CFURLCreateFromFSRef 32
- CFURLCreateResourcePropertiesForKeysFromBookmarkData 33
- CFURLCreateResourcePropertyForKeyFromBookmarkData 33
- CFURLCreateStringByAddingPercentEscapes 34
- CFURLCreateStringByReplacingPercentEscapes 35
- CFURLCreateStringByReplacingPercentEscapesUsingEncoding 36
- CFURLCreateWithBytes 37
- CFURLCreateWithFilePath 38
- CFURLCreateWithFilePathRelativeToBase 39
- CFURLCreateWithString 39
- CFURLGetBaseURL 40
- CFURLGetByteRangeForComponent 41
- CFURLGetBytes 41
- CFURLGetFileSystemRepresentation 42
- CFURLGetFSRef 43
- CFURLGetPortNumber 44
- CFURLGetString 44
- CFURLGetTypeID 45
- CFURLHasDirectoryPath 45
- CFURLResourceIsReachable 46
- CFURLSetResourcePropertiesForKeys 46
- CFURLSetResourcePropertyForKey 47
- CFURLSetTemporaryResourcePropertyForKey 47
- CFURLWriteBookmarkDataToFile 48
- Data Types 48
 - Bookmark Data Types 48
 - Miscellaneous 49
- Constants 50
 - Bookmark Data Constants 50
 - File System Constants 50
 - Miscellaneous 56

Document Revision History 61

Index 63

Listings

CFURL Reference 7

Listing 1 Code sample illustrating CFURLCopyLastPathComponent 15

CFURL Reference

Derived From:	CFType
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFURL.h

Overview

CFURL provides facilities for creating, parsing, and dereferencing URL strings. CFURL is useful to applications that need to use URLs to access resources, including local files.

A CFURL object is composed of two parts—a base URL, which can be `NULL`, and a string that is resolved relative to the base URL. A CFURL object whose string is fully resolved without a base URL is considered absolute; all others are considered relative.

CFURL fails to create an object if the string passed is not well-formed (that is, if it does not comply with RFC 2396). Examples of cases that will not succeed are strings containing space characters and high-bit characters. If a function fails to create a CFURL object, it returns `NULL`, which you must be prepared to handle. If you create CFURL objects using file system paths, you should use the [CFURLCreateFromFileSystemRepresentation](#) (page 31) and [CFURLCreateFromFileSystemRepresentationRelativeToBase](#) (page 31) functions, which handle the subtle differences between URL paths and file system paths.

For functions that read and write data from a URL, see *Core Foundation URL Access Utilities Reference*

CFURL is “toll-free bridged” with its Cocoa Foundation counterpart, NSURL. This means that the Core Foundation type is interchangeable in function or method calls with the bridged Foundation object. In other words, in a method where you see an `NSURL *` parameter, you can pass in a `CFURLRef`, and in a function where you see a `CFURLRef` parameter, you can pass in an NSURL instance. This also applies to concrete subclasses of NSURL. See *Integrating Carbon and Cocoa in Your Application* for more information on toll-free bridging.

Functions by Task

Creating a CFURL

[CFURLCopyAbsoluteURL](#) (page 12)

Creates a new CFURL object by resolving the relative portion of a URL against its base.

- [CFURLCreateAbsoluteURLWithBytes](#) (page 23)
Creates a new `CFURL` object by resolving the relative portion of a URL, specified as bytes, against its given base URL.
- [CFURLCreateByResolvingBookmarkData](#) (page 25)
Returns a new URL made by resolving bookmark data.
- [CFURLCreateCopyAppendingPathComponent](#) (page 26)
Creates a copy of a given URL and appends a path component.
- [CFURLCreateCopyAppendingPathExtension](#) (page 27)
Creates a copy of a given URL and appends a path extension.
- [CFURLCreateCopyDeletingLastPathComponent](#) (page 28)
Creates a copy of a given URL with the last path component deleted.
- [CFURLCreateCopyDeletingPathExtension](#) (page 28)
Creates a copy of a given URL with its last path extension removed.
- [CFURLCreateFilePathURL](#) (page 30)
Initializes and returns a newly created `CFURL` object as a file URL with a specified path.
- [CFURLCreateFileReferenceURL](#) (page 30)
Returns a new file reference URL that points to the same resource as a specified URL.
- [CFURLCreateFromFileSystemRepresentation](#) (page 31)
Creates a new `CFURL` object for a file system entity using the native representation.
- [CFURLCreateFromFileSystemRepresentationRelativeToBase](#) (page 31)
Creates a `CFURL` object from a native character string path relative to a base URL.
- [CFURLCreateFromFSRef](#) (page 32)
Creates a URL from a given directory or file.
- [CFURLCreateWithBytes](#) (page 37)
Creates a `CFURL` object using a given character bytes.
- [CFURLCreateWithFileSystemPath](#) (page 38)
Creates a `CFURL` object using a local file system path string.
- [CFURLCreateWithFileSystemPathRelativeToBase](#) (page 39)
Creates a `CFURL` object using a local file system path string relative to a base URL.
- [CFURLCreateWithString](#) (page 39)
Creates a `CFURL` object using a given `CFString` object.

Accessing the Parts of a URL

- [CFURLCanBeDecomposed](#) (page 11)
Determines if the given URL conforms to RFC 1808 and therefore can be decomposed.
- [CFURLCopyFileSystemPath](#) (page 12)
Returns the path portion of a given URL.
- [CFURLCopyFragment](#) (page 13)
Returns the fragment from a given URL.
- [CFURLCopyHostName](#) (page 14)
Returns the host name of a given URL.
- [CFURLCopyLastPathComponent](#) (page 14)
Returns the last path component of a given URL.

- [CFURLCopyNetLocation](#) (page 15)
Returns the net location portion of a given URL.
- [CFURLCopyParameterString](#) (page 16)
Returns the parameter string from a given URL.
- [CFURLCopyPassword](#) (page 16)
Returns the password of a given URL.
- [CFURLCopyPath](#) (page 17)
Returns the path portion of a given URL.
- [CFURLCopyPathExtension](#) (page 18)
Returns the path extension of a given URL.
- [CFURLCopyQueryString](#) (page 18)
Returns the query string of a given URL.
- [CFURLCopyResourceSpecifier](#) (page 20)
Returns any additional resource specifiers after the path.
- [CFURLCopyScheme](#) (page 21)
Returns the scheme portion of a given URL.
- [CFURLCopyStrictPath](#) (page 21)
Returns the path portion of a given URL.
- [CFURLCopyUserName](#) (page 22)
Returns the user name from a given URL.
- [CFURLGetPortNumber](#) (page 44)
Returns the port number from a given URL.
- [CFURLHasDirectoryPath](#) (page 45)
Determines if a given URL's path represents a directory.

Converting URLs to Other Representations

- [CFURLCreateData](#) (page 29)
Creates a `CFData` object containing the content of a given URL.
- [CFURLCreateStringByAddingPercentEscapes](#) (page 34)
Creates a copy of a string, replacing certain characters with the equivalent percent escape sequence based on the specified encoding.
- [CFURLCreateStringByReplacingPercentEscapes](#) (page 35)
Creates a new string by replacing any percent escape sequences with their character equivalent.
- [CFURLCreateStringByReplacingPercentEscapesUsingEncoding](#) (page 36)
Creates a new string by replacing any percent escape sequences with their character equivalent.
- [CFURLGetFileSystemRepresentation](#) (page 42)
Fills a buffer with the file system's native string representation of a given URL's path.
- [CFURLGetFSRef](#) (page 43)
Converts a given URL to a file or directory object.
- [CFURLGetString](#) (page 44)
Returns the URL as a `CFString` object.

Getting URL Properties

[CFURLGetBaseURL](#) (page 40)

Returns the base URL of a given URL if it exists.

[CFURLGetBytes](#) (page 41)

Returns by reference the byte representation of a URL object.

[CFURLGetByteRangeForComponent](#) (page 41)

Returns the range of the specified component in the bytes of a URL.

[CFURLGetTypeID](#) (page 45)

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

[CFURLResourceIsReachable](#) (page 46)

Returns whether the resource pointed to by a file URL can be reached.

Getting and Setting File System Resource Properties

[CFURLClearResourcePropertyCache](#) (page 11)

Clears all cached resource property values of a given URL.

[CFURLClearResourcePropertyCacheForKey](#) (page 12)

Discards a cached property value for a given key of a given URL.

[CFURLCopyResourcePropertiesForKeys](#) (page 19)

Returns any number of resource property values of a URL as a dictionary.

[CFURLCopyResourcePropertyForKey](#) (page 20)

Returns the value of a given resource property of a given URL.

[CFURLCreateResourcePropertiesForKeysFromBookmarkData](#) (page 33)

Returns the resource values for properties identified by a specified array of keys contained in specified bookmark data.

[CFURLCreateResourcePropertyForKeyFromBookmarkData](#) (page 33)

Returns the value of a resource property from specified bookmark data.

[CFURLSetResourcePropertiesForKeys](#) (page 46)

Sets resource properties of a URL specified by a given dictionary of keys and values.

[CFURLSetResourcePropertyForKey](#) (page 47)

Sets the resource property of the URL specified by a given key to a given value.

[CFURLSetTemporaryResourcePropertyForKey](#) (page 47)

Sets the resource property of the URL specified by a given key to a given value without writing the assignment to the backing store.

Working with Bookmark Data

[CFURLCreateBookmarkData](#) (page 23)

Returns bookmark data for a URL, created with specified options and resource values.

[CFURLCreateBookmarkDataFromAliasRecord](#) (page 24)

Initializes and returns bookmark data derived from an alias record.

[CFURLCreateBookmarkDataFromFile](#) (page 25)

Initializes and returns bookmark data derived from a file pointed to by a specified URL.

[CFURLWriteBookmarkDataToFile](#) (page 48)

Creates an alias file on disk at a specified location with specified bookmark data.

Functions

CFURLCanBeDecomposed

Determines if the given URL conforms to RFC 1808 and therefore can be decomposed.

```
Boolean CFURLCanBeDecomposed (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to test.

Return Value

true if *anURL* conforms to RFC 1808, false otherwise.

Discussion

If a CFURL object can be decomposed, you can retrieve separately each of the four components (scheme, net location, path, and resource specifier), as well as the base URL.

Relative URLs are permitted to have only paths (or a variety of other configurations); these are considered decomposable if their base URL is decomposable. If no base URL is present, they are considered decomposable.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFURL.h

CFURLClearResourcePropertyCache

Clears all cached resource property values of a given URL.

```
void CFURLClearResourcePropertyCache (
    CFURLRef url
);
```

Parameters

url

The URL.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLClearResourcePropertyCacheForKey

Discards a cached property value for a given key of a given URL.

```
void CFURLClearResourcePropertyCacheForKey (
    CFURLRef url,
    CFStringRef key
);
```

Parameters

url

The URL.

key

The property value key.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCopyAbsoluteURL

Creates a new CFURL object by resolving the relative portion of a URL against its base.

```
CFURLRef CFURLCopyAbsoluteURL (
    CFURLRef relativeURL
);
```

Parameters

relativeURL

The CFURL object to resolve.

Return Value

A new CFURL object, or NULL if *relativeURL* cannot be made absolute. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

ImageClient

Declared In

CFURL.h

CFURLCopyFileSystemPath

Returns the path portion of a given URL.

```
CFStringRef CFURLCopyFilePath (
    CFURLRef anURL,
    CFURLPathStyle pathStyle
);
```

Parameters*anURL*

The CFURL object whose path you want to obtain.

pathStyle

The operating system path style to be used to create the path. See [Path Style](#) (page 58) for a list of possible values.

Return Value

The URL's path in the format specified by *pathStyle*. Ownership follows the Create Rule.

Discussion

This function returns the URL's path as a file system path for a given path style.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

AudioBurn

DisplayURL

FinalCutPro_AppleEvents

MoreAppleEvents

SeeMyFriends

Declared In

CFURL.h

CFURLCopyFragment

Returns the fragment from a given URL.

```
CFStringRef CFURLCopyFragment (
    CFURLRef anURL,
    CFStringRef charactersToLeaveEscaped
);
```

Parameters*anURL*

The CFURL object whose fragment you want to obtain.

charactersToLeaveEscaped

Characters whose percent escape sequences, such as %20 for a space character, you want to leave intact. Pass NULL to specify that no percent escapes be replaced, or the empty string (CFSTR(" ")) to specify that all be replaced.

Return Value

The fragment, or NULL if no fragment exists. Ownership follows the Create Rule.

Discussion

A fragment is the text following a "#". These are generally used to indicate locations within a single file. This function removes all percent escape sequences except those for characters specified in *charactersToLeaveEscaped*.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyHostName

Returns the host name of a given URL.

```
CFStringRef CFURLCopyHostName (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to examine.

Return Value

The host name of *anURL*. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

ImageClient

Declared In

CFURL.h

CFURLCopyLastPathComponent

Returns the last path component of a given URL.

```
CFStringRef CFURLCopyLastPathComponent (
    CFURLRef url
);
```

Parameters

url

The CFURL object to examine.

Return Value

The last path component of *url*. Ownership follows the Create Rule.

Discussion

Note that if there is no last path component, this function returns an empty string. In the code sample shown in Listing 1, `lastPathComponent` is an empty string.

Listing 1 Code sample illustrating `CFURLCopyLastPathComponent`

```
CFStringRef urlString = CFSTR("http://www.apple.com");
CFURLRef url = CFURLCreateWithString(NULL, urlString, NULL);
CFStringRef lastPathComponent = CFURLCopyLastPathComponent(url);
```

If `urlString` were created with `CFSTR("http://www.apple.com/")`, then `lastPathComponent` would be a `CFString` object containing the character `"/"`.

See also [CFURLCopyPathExtension](#) (page 18).

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

CFFTPSample

DisplayURL

OpenALExample

RecentItems

SimpleAudioExtraction

Declared In

`CFURL.h`

CFURLCopyNetLocation

Returns the net location portion of a given URL.

```
CFStringRef CFURLCopyNetLocation (
    CFURLRef anURL
);
```

Parameters

anURL

The `CFURL` object to examine.

Return Value

The net location of *anURL*, or `NULL` if the URL cannot be decomposed (doesn't conform to RFC 1808). Ownership follows the Create Rule.

Discussion

The URL net location is the portion of the URL that identifies the network address of the resource. It includes the optional username and password, as well as the target machine's IP address or host name.

This function leaves any percent escape sequences intact.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyParameterString

Returns the parameter string from a given URL.

```
CFStringRef CFURLCopyParameterString (
    CFURLRef anURL,
    CFStringRef charactersToLeaveEscaped
);
```

Parameters

anURL

The CFURL object to examine.

charactersToLeaveEscaped

Characters whose percent escape sequences, such as %20 for a space character, you want to leave intact. Pass NULL to specify that no percent escapes be replaced, or the empty string (CFSTR("")) to specify that all be replaced.

Return Value

The parameter string (as defined in RFC 1738), or NULL if no parameter string exists. Ownership follows the Create Rule.

Discussion

This function removes all percent escape sequences except those for characters specified in *charactersToLeaveEscaped*.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyPassword

Returns the password of a given URL.

```
CFStringRef CFURLCopyPassword (
    CFURLRef anURL
);
```

Parameters*anURL*

The CFURL object to examine.

Return Value

The password, or NULL if no password exists. In some cases, this function may also return the empty string (CFSTR("")) if no password exists. You should consider NULL and the empty string to be equivalent. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyPath

Returns the path portion of a given URL.

```
CFStringRef CFURLCopyPath (
    CFURLRef anURL
);
```

Parameters*anURL*

The CFURL object to examine.

Return Value

The path of *anURL*, or NULL if the URL cannot be decomposed (doesn't conform to RFC 1808). Ownership follows the Create Rule.

Discussion

This function does not resolve the URL against its base and replaces all percent escape sequences. This function's return value includes any leading slash (giving the path the normal POSIX appearance), if present. If this behavior is not appropriate, use [CFURLCopyStrictPath](#) (page 21) whose return value omits any leading slash. You may also want to use the function [CFURLCopyFilePath](#) (page 12), which returns the URL's path as a file system path for the given path style. If the path is to be passed to file system calls, you may also want to use the function [CFURLGetFileSystemRepresentation](#) (page 42), which returns a C string.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

ConvertFile

DisplayURL

ImageClient
SampleDS

Declared In
CFURL.h

CFURLCopyPathExtension

Returns the path extension of a given URL.

```
CFStringRef CFURLCopyPathExtension (
    CFURLRef url
);
```

Parameters

url

The CFURL object to examine.

Return Value

The path extension of *url*, or NULL if no extension exists. Ownership follows the Create Rule.

Discussion

The path extension is the portion of the last path component which follows the final period, if there is one. For example, for `http://www.apple.com/developer/macosx.today.html`, the extension is `html`, and for `http://www.apple.com/developer`, there is no path extension.

See also [CFURLCopyLastPathComponent](#) (page 14).

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyQueryString

Returns the query string of a given URL.

```
CFStringRef CFURLCopyQueryString (
    CFURLRef anURL,
    CFStringRef charactersToLeaveEscaped
);
```

Parameters

anURL

The CFURL object to examine.

charactersToLeaveEscaped

Characters whose percent escape sequences, such as %20 for a space character, you want to leave intact. Pass `NULL` to specify that no percent escapes be replaced, or the empty string (`CFSTR("")`) to specify that all be replaced.

Return Value

The query string, or `NULL` if no parameter string exists. Ownership follows the Create Rule.

Discussion

This function removes all percent escape sequences except those for characters specified in *charactersToLeaveEscaped*.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyResourcePropertiesForKeys

Returns any number of resource property values of a URL as a dictionary.

```
CFDictionaryRef CFURLCopyResourcePropertiesForKeys (
    CFURLRef url,
    CFArrayRef keys,
    CFErrorRef *error
);
```

Parameters

url

The URL.

keys

The property value keys that values are requested for.

error

The error that occurred in the case that the bookmark data cannot be created.

Return Value

A dictionary of resource property values, or `NULL` if an error occurs.

Discussion

Values are fetched synchronously from the resource backing store unless they are already cached.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCopyResourcePropertyForKey

Returns the value of a given resource property of a given URL.

```

Boolean CFURLCopyResourcePropertyForKey (
    CFURLRef url,
    CFStringRef key,
    void *propertyValueTypeRefPtr,
    CFErrorRef *error
);

```

Parameters

url

The URL.

key

The property value key that the value is requested for.

propertyValueTypeRefPtr

The pointer that is populated with the result.

error

The error that occurred in the case that the bookmark data cannot be created.

Return Value

true if *propertyValueTypeRefPtr* is successfully populated; otherwise, false.

Discussion

Values are fetched synchronously from the resource backing store unless they are already cached.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCopyResourceSpecifier

Returns any additional resource specifiers after the path.

```

CFStringRef CFURLCopyResourceSpecifier (
    CFURLRef anURL
);

```

Parameters

anURL

The CFURL object to examine.

Return Value

The resource specifiers. Ownership follows the Create Rule.

Discussion

This function leaves any percent escape sequences intact. For decomposable URLs, this function returns everything after the path. For URLs that cannot be decomposed, this function returns everything except the scheme itself.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyScheme

Returns the scheme portion of a given URL.

```
CFStringRef CFURLCopyScheme (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to examine.

Return Value

The scheme of *anURL*. Ownership follows the Create Rule.

Discussion

The URL scheme is the portion of the URL specifying the transport type. For example `http`, `ftp`, and `rtsp` are schemes. This function leaves any percent escape sequences intact.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

ImageClient

Declared In

CFURL.h

CFURLCopyStrictPath

Returns the path portion of a given URL.

```
CFStringRef CFURLCopyStrictPath (
    CFURLRef anURL,
    Boolean *isAbsolute
);
```

Parameters

anURL

The CFURL object to examine.

isAbsolute

On return, indicates whether the path of *anURL* is absolute.

Return Value

The path of *anURL*, or `NULL` if the URL cannot be decomposed (doesn't conform to RFC 1808). Ownership follows the Create Rule.

Discussion

This function does not resolve the URL against its base and replaces all percent escape sequences. This function's return value does not include a leading slash and uses *isAbsolute* to report whether the URL's path is absolute. If this behavior is not appropriate, use the [CFURLCopyPath](#) (page 17) function whose return value includes the leading slash (giving the path the normal POSIX appearance). You may also want to use the [CFURLCopyFilePath](#) (page 12) function, which returns the URL's path as a file system path for the given path style. If the path is to be passed to file system calls, you may also want to use the function [CFURLGetFileSystemRepresentation](#) (page 42), which returns a C string.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCopyUserName

Returns the user name from a given URL.

```
CFStringRef CFURLCopyUserName (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to examine.

Return Value

The user name, or `NULL` if no user name exists. In some cases, this function may also return the empty string (`CFSTR("")`) if no username exists. You should consider `NULL` and the empty string to be equivalent. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCreateAbsoluteURLWithBytes

Creates a new CFURL object by resolving the relative portion of a URL, specified as bytes, against its given base URL.

```
CFURLRef CFURLCreateAbsoluteURLWithBytes (
    CFAllocatorRef alloc,
    const UInt8 *relativeURLBytes,
    CFIndex length,
    CFStringEncoding encoding,
    CFURLRef baseURL,
    Boolean useCompatibilityMode
);
```

Parameters

allocator

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

relativeURLBytes

The character bytes that represent a relative URL to convert into a CFURL object.

length

The number of bytes in *relativeURLBytes*.

encoding

The string encoding of the *relativeURLBytes* string. This encoding is also used to interpret percent escape sequences.

baseURL

The URL to which *relativeURLBytes* is relative.

useCompatibilityMode

If true, the rules historically used on the web are used to resolve the string specified by the *relativeURLBytes* parameter against *baseURL*. These rules are generally listed in the RFC as optional or alternate interpretations. Otherwise, the strict rules from the RFC are used.

Return Value

A new CFURL object, or NULL if *relativeURLBytes* cannot be made absolute. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.3 and later.

Declared In

CFURL.h

CFURLCreateBookmarkData

Returns bookmark data for a URL, created with specified options and resource values.

```
CFDataRef CFURLCreateBookmarkData (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFURLBookmarkCreationOptions options,
    CFArrayRef resourcePropertiesToInclude,
    CFURLRef relativeToURL,
    CFErrorRef *error
);
```

Parameters*allocator*

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

url

The URL that bookmark data is being created for.

options

Options taken into account when creating the bookmark data.

resourcePropertiesToInclude

An array of names of URL resource properties.

relativeToURL

The URL that the bookmark data is relative to.

error

The error that occurred in the case that the bookmark data cannot be created.

Return Value

The bookmark data for the URL.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateBookmarkDataFromAliasRecord

Initializes and returns bookmark data derived from an alias record.

```
CFDataRef CFURLCreateBookmarkDataFromAliasRecord (
    CFAllocatorRef allocatorRef,
    CFDataRef aliasRecordDataRef
);
```

Parameters*allocatorRef*

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

aliasRecordDataRef

The alias record.

Return Value

The bookmark data for the alias record.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateBookmarkDataFromFile

Initializes and returns bookmark data derived from a file pointed to by a specified URL.

```
CFDataRef CFURLCreateBookmarkDataFromFile (
    CFAllocatorRef allocator,
    CFURLRef fileURL,
    CFErrorRef *errorRef
);
```

Parameters

allocator

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

fileURL

The file URL.

errorRef

The error that occurred in the case that the bookmark data cannot be created.

Return Value

The bookmark data for the file, or `NULL` if an error occurs.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateByResolvingBookmarkData

Returns a new URL made by resolving bookmark data.

```
CFURLRef CFURLCreateByResolvingBookmarkData (
    CFAllocatorRef allocator,
    CFDataRef bookmark,
    CFURLBookmarkResolutionOptions options,
    CFURLRef relativeToURL,
    CFArrayRef resourcePropertiesToInclude,
    Boolean *isStale,
    CFErrorRef *error
);
```

Parameters

allocator

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

bookmark

The bookmark data the URL is derived from.

options

Options taken into account when resolving the bookmark data.

relativeToURL

The base URL that the bookmark data is relative to. Can be NULL.

resourcePropertiesToInclude

An array of resource properties to include when creating the URL. Can be NULL.

isStale

If YES, the bookmark data is stale.

error

The error that occurred in the case that the URL cannot be created.

Return Value

A new URL made by resolving *bookmark*, or NULL if an error occurs.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateCopyAppendingPathComponent

Creates a copy of a given URL and appends a path component.

```
CFURLRef CFURLCreateCopyAppendingPathComponent (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFStringRef pathComponent,
    Boolean isDirectory
);
```

Parameters

allocator

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

url

The CFURL object to which to append a path component.

pathComponent

The path component to append to *url*.

isDirectory

A Boolean value that specifies whether the string is treated as a directory path when resolving against relative path components. Pass `true` if the new component indicates a directory, `false` otherwise.

Return Value

A copy of *url* appended with *pathComponent*. Ownership follows the Create Rule.

Discussion

The *isDirectory* argument specifies whether or not the new path component points to a file or a directory. Note that the URL syntax for a directory and for a file at otherwise the same location are slightly different—directory URLs must end in “/”. If you have the URL `http://www.apple.com/foo/` and you append the path component `bar`, then if *isDirectory* is YES then the resulting URL is `http://www.apple.com/foo/bar/`, whereas if *isDirectory* is NO then the resulting URL is `http://www.apple.com/foo/bar`. This difference is particularly important if you resolve another URL against this new URL. `file.html` relative to `http://www.apple.com/foo/bar` is `http://www.apple.com/foo/file.html`, whereas `file.html` relative to `http://www.apple.com/foo/bar/` is `http://www.apple.com/foo/bar/file.html`.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

CFFTPSample

simpleJavaLauncher

SpellingChecker CarbonCocoa Bundled

SpellingChecker-CarbonCocoa

Declared In

CFURL.h

CFURLCreateCopyAppendingPathExtension

Creates a copy of a given URL and appends a path extension.

```
CFURLRef CFURLCreateCopyAppendingPathExtension (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFStringRef extension
);
```

Parameters

allocator

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

url

The CFURL object to which to append a path extension.

extension

The extension to append to *url*.

Return Value

A copy of *url* appended with *extension*. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLCreateCopyDeletingLastPathComponent

Creates a copy of a given URL with the last path component deleted.

```
CFURLRef CFURLCreateCopyDeletingLastPathComponent (
    CFAllocatorRef allocator,
    CFURLRef url
);
```

Parameters*allocator*

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

url

The CFURL object whose last path component you want to delete.

Return Value

A copy of *url* with the last path component deleted. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

HID Utilities Source

ImageClient

Declared In

CFURL.h

CFURLCreateCopyDeletingPathExtension

Creates a copy of a given URL with its last path extension removed.

```
CFURLRef CFURLCreateCopyDeletingPathExtension (
    CFAllocatorRef allocator,
    CFURLRef url
);
```

Parameters*allocator*

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

url

The CFURL object whose path extension you want to delete.

Return Value

A copy of *url* with its last path extension removed. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

OpenALExample

Declared In

CFURL.h

CFURLCreateData

Creates a `CFData` object containing the content of a given URL.

```
CFDataRef CFURLCreateData (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFStringEncoding encoding,
    Boolean escapeWhitespace
);
```

Parameters

allocator

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

url

The URL to convert into a `CFData` object.

encoding

The string encoding to use when converting *url* into a `CFData` object.

escapeWhitespace

`true` if you want to escape whitespace characters in the URL, `false` otherwise.

Return Value

A new `CFData` object containing the content of *url*. Ownership follows the Create Rule.

Discussion

This function escapes any character that is not 7-bit ASCII with the byte-code for the given encoding. If *escapeWhitespace* is `true`, whitespace characters (' ', '\t', '\r', '\n') will be escaped as well. This is desirable if you want to embed the URL into a larger text stream like HTML.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFURL.h

CFURLCreateFilePathURL

Initializes and returns a newly created CFURL object as a file URL with a specified path.

```
CFURLRef CFURLCreateFilePathURL (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFErrorRef *error
);
```

Parameters

allocator

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

url

The path.

error

The error that occurred in the case that the URL cannot be created.

Return Value

A CFURL object initialized with *url*, or NULL if an error occurs.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateFileReferenceURL

Returns a new file reference URL that points to the same resource as a specified URL.

```
CFURLRef CFURLCreateFileReferenceURL (
    CFAllocatorRef allocator,
    CFURLRef url,
    CFErrorRef *error
);
```

Parameters

allocator

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

url

The URL.

error

The error that occurred in the case that the URL cannot be created.

Return Value

The new file reference URL, or NULL if an error occurs.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateFromFileSystemRepresentation

Creates a new CFURL object for a file system entity using the native representation.

```
CFURLRef CFURLCreateFromFileSystemRepresentation (
    CFAllocatorRef allocator,
    const UInt8 *buffer,
    CFIndex bufLen,
    Boolean isDirectory
);
```

Parameters*allocator*

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

buffer

The character bytes to convert into a CFURL object. This should be the path as you would use in POSIX function calls.

bufLen

The number of bytes in the buffer.

isDirectory

A Boolean value that specifies whether the string is treated as a directory path when resolving against relative path components—`true` if the pathname indicates a directory, `false` otherwise.

Return Value

A new CFURL object. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

AudioQueueTools

CFFTPSample

ConvertFile

MemoryBasedBundle

OpenCL NBody Simulation Example

Declared In

CFURL.h

CFURLCreateFromFileSystemRepresentationRelativeToBase

Creates a CFURL object from a native character string path relative to a base URL.

```
CFURLRef CFURLCreateFromFileSystemRepresentationRelativeToBase (
    CFAllocatorRef allocator,
    const UInt8 *buffer,
    CFIndex bufLen,
    Boolean isDirectory,
    CFURLRef baseURL
);
```

Parameters*allocator*

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

buffer

The character bytes to convert into a CFURL object. This should be the path as you would use in POSIX function calls.

bufLen

The number of bytes in the buffer.

isDirectory

A Boolean value that specifies whether the string is treated as a directory path when resolving against relative path components. Pass `true` if the pathname indicates a directory, `false` otherwise.

baseURL

The URL against which to resolve the path.

Return Value

A new CFURL object. Ownership follows the Create Rule.

Discussion

This function takes a path name in the form of a native character string, resolves it against a base URL, and returns a new CFURL object containing the result.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFURL.h

CFURLCreateFromFSRef

Creates a URL from a given directory or file.

```
CFURLRef CFURLCreateFromFSRef (
    CFAllocatorRef allocator,
    const struct FSRef *fsRef
);
```

Parameters*allocator*

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

fsRef

The file or directory representing the URL.

Return Value

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

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

MoreSCF

SeeMyFriends

SimpleAudioExtraction

SimpleScriptingPlugin

Declared In

`CFURL.h`

CFURLCreateResourcePropertiesForKeysFromBookmarkData

Returns the resource values for properties identified by a specified array of keys contained in specified bookmark data.

```
CFDictionaryRef CFURLCreateResourcePropertiesForKeysFromBookmarkData (
    CFAllocatorRef allocator,
    CFArrayRef resourcePropertiesToReturn,
    CFDataRef bookmark
);
```

Parameters

allocator

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

resourcePropertiesToReturn

An array of names of URL resource properties.

bookmark

The bookmark data the resource values are derived from.

Return Value

A dictionary of the requested resource values contained in *bookmarkData*.

Discussion

This function does not attempt to resolve the bookmark data or perform I/O.

Availability

Available in Mac OS X v10.6 and later.

Declared In

`CFURL.h`

CFURLCreateResourcePropertyForKeyFromBookmarkData

Returns the value of a resource property from specified bookmark data.

```

CFTypeRef CFURLCreateResourcePropertyForKeyFromBookmarkData (
    CFAllocatorRef allocator,
    CFStringRef resourcePropertyKey,
    CFDataRef bookmark
);

```

Parameters*allocator*

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

resourcePropertyKey

The resource property key.

bookmark

The bookmark data the resource value is derived from.

Return Value

The resource property value.

Discussion

This function does not attempt to resolve the bookmark data or perform I/O.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLCreateStringByAddingPercentEscapes

Creates a copy of a string, replacing certain characters with the equivalent percent escape sequence based on the specified encoding.

```

CFStringRef CFURLCreateStringByAddingPercentEscapes (
    CFAllocatorRef allocator,
    CFStringRef originalString,
    CFStringRef charactersToLeaveUnescaped,
    CFStringRef legalURLCharactersToBeEscaped,
    CFStringEncoding encoding
);

```

Parameters*allocator*

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

originalString

The CFString object to copy.

charactersToLeaveUnescaped

Characters whose percent escape sequences you want to leave intact. Pass NULL to specify that all escape sequences be replaced.

legalURLCharactersToBeEscaped

Legal characters to be escaped. Pass NULL to specify that no legal characters be replaced.

encoding

The encoding to use for the translation. If you are uncertain of the correct encoding, you should use UTF-8, which is the encoding designated by RFC 2396 as the correct encoding for use in URLs.

Return Value

A copy of *originalString* replacing certain characters. If it does not need to be modified (no percent escape sequences are missing), this function may merely return *originalString* with its reference count incremented. Ownership follows the Create Rule.

Discussion

The characters escaped are all characters that are not legal URL characters (based on RFC 2396), plus any characters in *legalURLCharactersToBeEscaped*, less any characters in *charactersToLeaveUnescaped*. To simply correct any non-URL characters in an otherwise correct URL string, pass NULL for the *allocator*, *charactersToLeaveEscaped*, and *legalURLCharactersToBeEscaped* parameters, and *kCFStringEncodingUTF8* as the *encoding* parameter.

It may be difficult to use this function to "clean up" unescaped or partially escaped URL strings where sequences are unpredictable and you cannot specify *charactersToLeaveUnescaped*. Instead, you can "pre-process" a URL string using [CFURLCreateStringByReplacingPercentEscapesUsingEncoding](#) (page 36) then add the escape characters using [CFURLCreateStringByAddingPercentEscapes](#) (page 34), as shown in the following code fragment.

```
CFStringRef originalURLString =
CFSTR("http://online.store.com/storefront/?request=get-document&doi=10.1175%2F1520-0426(2005)014%3C1157:DDADSS%3E2.0.CO%3B2");
CFStringRef preprocessedString =
    CFURLCreateStringByReplacingPercentEscapesUsingEncoding(kCFAllocatorDefault,
        originalURLString, CFSTR(""), kCFStringEncodingUTF8);
CFStringRef urlString =
    CFURLCreateStringByAddingPercentEscapes(kCFAllocatorDefault,
        preprocessedString, NULL, NULL, kCFStringEncodingUTF8);
url = CFURLCreateWithString(kCFAllocatorDefault, urlString, NULL);
```

Availability

Available in CarbonLib v1.3 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

CFNetworkHTTPDownload

Declared In

CFURL.h

CFURLCreateStringByReplacingPercentEscapes

Creates a new string by replacing any percent escape sequences with their character equivalent.

```
CFStringRef CFURLCreateStringByReplacingPercentEscapes (
    CFAllocatorRef allocator,
    CFStringRef originalString,
    CFStringRef charactersToLeaveEscaped
);
```

Parameters*allocator*

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

originalString

The `CFString` object to be copied and modified.

charactersToLeaveEscaped

Characters whose percent escape sequences, such as `%20` for a space character, you want to leave intact. Pass `NULL` to specify that no percent escapes be replaced, or the empty string (`CFSTR(" ")`) to specify that all be replaced.

Return Value

A new `CFString` object, or `NULL` if the percent escapes cannot be converted to characters, assuming UTF-8 encoding. If no characters need to be replaced, this function returns the original string with its reference count incremented. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

`CFNetworkHTTPDownload`

Declared In

`CFURL.h`

CFURLCreateStringByReplacingPercentEscapesUsingEncoding

Creates a new string by replacing any percent escape sequences with their character equivalent.

```
CFStringRef CFURLCreateStringByReplacingPercentEscapesUsingEncoding (
    CFAllocatorRef allocator,
    CFStringRef origString,
    CFStringRef charsToLeaveEscaped,
    CFStringEncoding encoding
);
```

Parameters*allocator*

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

originalString

The `CFString` object to be copied and modified.

charactersToLeaveEscaped

Characters whose percent escape sequences, such as %20 for a space character, you want to leave intact. Pass `NULL` to specify that no percent escapes be replaced, or the empty string (`CFSTR(" ")`) to specify that all be replaced.

encoding

Specifies the encoding to use when interpreting percent escapes.

Return Value

A new `CFString` object, or `NULL` if the percent escapes cannot be converted to characters, assuming the encoding given by *encoding*. If no characters need to be replaced, this function returns the original string with its reference count incremented. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`CFURL.h`

CFURLCreateWithBytes

Creates a `CFURL` object using a given character bytes.

```
CFURLRef CFURLCreateWithBytes (
    CFAllocatorRef allocator,
    const UInt8 *URLBytes,
    CFIndex length,
    CFStringEncoding encoding,
    CFURLRef baseURL
);
```

Parameters*allocator*

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

URLBytes

The character bytes to convert into a `CFURL` object.

length

The number of bytes in *URLBytes*.

encoding

The string encoding of the *URLBytes* string. This encoding is also used to interpret percent escape sequences.

baseURL

The URL to which *URLBytes* is relative. Pass `NULL` if *URLBytes* contains an absolute URL or if you want to create a relative URL. If *URLBytes* contains an absolute URL, this parameter is ignored.

Return Value

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

Discussion

The specified string encoding will be used both to interpret *URLBytes*, and to interpret any percent-escapes within the string.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

RecentItems

Declared In

CFURL.h

CFURLCreateWithFilePath

Creates a CFURL object using a local file system path string.

```
CFURLRef CFURLCreateWithFilePath (
    CFAllocatorRef allocator,
    CFStringRef filePath,
    CFURLPathStyle pathStyle,
    Boolean isDirectory
);
```

Parameters

allocator

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

filePath

The path string to convert to a CFURL object.

pathStyle

The operating system path style used in *filePath*. See [Path Style](#) (page 58) for a list of possible values.

isDirectory

A Boolean value that specifies whether *filePath* is treated as a directory path when resolving against relative path components. Pass `true` if the pathname indicates a directory, `false` otherwise.

Return Value

A new CFURL object. Ownership follows the Create Rule.

Discussion

If *filePath* is not absolute, the resulting URL will be considered relative to the current working directory (evaluated when this function is being invoked).

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

From A View to A Movie

From A View to A Picture

Quartz EB

TexturePerformanceDemo

Declared In

CFURL.h

CFURLCreateWithFileSystemPathRelativeToBase

Creates a CFURL object using a local file system path string relative to a base URL.

```
CFURLRef CFURLCreateWithFileSystemPathRelativeToBase (
    CFAllocatorRef allocator,
    CFStringRef filePath,
    CFURLPathStyle pathStyle,
    Boolean isDirectory,
    CFURLRef baseURL
);
```

Parameters*allocator*

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

filePath

The path string to convert to a CFURL object.

pathStyle

The operating system path style used in the *filePath* string. See [Path Style](#) (page 58) for a list of possible values.

isDirectory

A Boolean value that specifies whether *filePath* is treated as a directory path when resolving against relative path components. Pass `true` if the pathname indicates a directory, `false` otherwise.

baseURL

The base URL against which to resolve the *filePath*.

Return Value

A new CFURL object. Ownership follows the Create Rule.

Discussion

This function takes a path name in the form of a CFString object, resolves it against a base URL, and returns a new CFURL object containing the result.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

Aperture Image Resizer

DisplayURL

Declared In

CFURL.h

CFURLCreateWithString

Creates a CFURL object using a given CFString object.

```
CFURLRef CFURLCreateWithString (
    CFAllocatorRef allocator,
    CFStringRef urlString,
    CFURLRef baseURL
);
```

Parameters*allocator*

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

URLString

The CFString object containing the URL string.

baseURL

The URL to which *URLString* is relative. Pass `NULL` if *URLString* contains an absolute URL or if you want to create a relative URL. If *URLString* contains an absolute URL, *baseURL* is ignored.

Return Value

A new CFURL object. Ownership follows the Create Rule.

Discussion

Any escape sequences in *URLString* will be interpreted using UTF-8.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

AuthForAll

CFFTPSample

DisplayURL

DockBrowser

OpenALExample

Declared In

CFURL.h

CFURLGetBaseURL

Returns the base URL of a given URL if it exists.

```
CFURLRef CFURLGetBaseURL (
    CFURLRef anURL
);
```

Parameters*anURL*

The CFURL object to examine.

Return Value

A CFURL object representing the base URL of *anURL*. Ownership follows the Get Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLGetByteRangeForComponent

Returns the range of the specified component in the bytes of a URL.

```
CFRange CFURLGetByteRangeForComponent (
    CFURLRef url,
    CFURLComponentType component,
    CFRange *rangeIncludingSeparators
);
```

Parameters

anURL

The URL containing *component*.

component

The type of component in *anURL* whose range you want to obtain. See [Component Type](#) (page 56) for possible values.

rangeIncludingSeparators

Specifies the range of *component* including the sequences that separate component from the previous and next components. If there is no previous or next components, this function will match the range of the component itself. If *anURL* does not contain *component*, *rangeIncludingSeparators* is set to the location where the component would be inserted.

Return Value

The range of bytes for *component* in the buffer returned by the [CFURLGetBytes](#) (page 41) function. If *anURL* does not contain *component*, the first part of the returned range is set to `kCFNotFound`.

Discussion

This function is intended to be used in conjunction with the [CFURLGetBytes](#) (page 41) function, since the range returned is only applicable to the bytes returned by [CFURLGetBytes](#) (page 41).

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLGetBytes

Returns by reference the byte representation of a URL object.

```
CFIndex CFURLGetBytes (
    CFURLRef url,
    UInt8 *buffer,
    CFIndex bufferLength
);
```

Parameters*anURL*

The URL object to convert to a byte representation.

buffer

The buffer where you want the bytes to be placed. If the buffer is of insufficient size, returns -1 and no bytes are placed in buffer. If NULL the needed length is computed and returned. The returned bytes are the original bytes from which the URL was created. If the URL was created from a string, the bytes are the bytes of the string encoded via UTF-8.

bufferLength

The number of bytes in *buffer*.

Return Value

Returns the number of bytes in *buffer* that were filled. If the buffer is of insufficient size, returns -1.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

DisplayURL

Declared In

CFURL.h

CFURLGetFileSystemRepresentation

Fills a buffer with the file system's native string representation of a given URL's path.

```
Boolean CFURLGetFileSystemRepresentation (
    CFURLRef url,
    Boolean resolveAgainstBase,
    UInt8 *buffer,
    CFIndex maxBufLen
);
```

Parameters*url*

The CFURL object whose native file system representation you want to obtain.

resolveAgainstBase

Pass true to return an absolute path name.

buffer

A pointer to a character buffer. On return, the buffer holds the native file system's representation of *url*. The buffer is null-terminated. This parameter must be at least *maxBufLen* in size for the file system in question to avoid failures for insufficiently large buffers.

maxBufLen

The maximum number of characters that can be written to *buffer*.

Return Value

`true` if successful, `false` if an error occurred.

Discussion

No more than `maxBufLen` bytes are written to `buffer`. If `url` requires more than `maxBufLen` bytes to represent itself, including the terminating null byte, this function returns `false`. To avoid this possible failure, you should pass a buffer with size of at least the maximum path length for the file system in question.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BetterAuthorizationSample

BSDLLCTest

DisplayURL

EmbeddedAppleScripts

MemoryBasedBundle

Declared In

CFURL.h

CFURLGetFSRef

Converts a given URL to a file or directory object.

```
Boolean CFURLGetFSRef (
    CFURLRef url,
    struct FSRef *fsRef
);
```

Parameters

url

The CFURL object to convert to a file or directory object.

fsRef

Upon return, contains the file or directory object representing *url*.

Return Value

`true` if the conversion was successful, otherwise `false`.

Special Considerations

The function cannot create an FSRef object if the path specified by *url* contains an alias. The function can, however, traverse symbolic links.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BasicInputMethod

CoreTextTest

QTGraphicsImport

TexturePerformanceDemo
TextureRange

Declared In
CFURL.h

CFURLGetPortNumber

Returns the port number from a given URL.

```
SInt32 CFURLGetPortNumber (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to examine.

Return Value

The port number of *anURL*, or -1 if no port number exists.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

DisplayURL

ImageClient

Declared In

CFURL.h

CFURLGetString

Returns the URL as a CFString object.

```
CFStringRef CFURLGetString (
    CFURLRef anURL
);
```

Parameters

anURL

The CFURL object to convert into a CFString object.

Return Value

A string representation of *anURL*. Ownership follows the Get Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

AlbumToSlideshow

DisplayURL
FSMegaInfo
LoginItemsAE

Declared In

CFURL.h

CFURLGetTypeID

Returns the type identifier for the CFURL opaque type.

```
CTypeID CFURLGetTypeID (  
    void  
);
```

Return Value

The type identifier for the CFURL opaque type.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

LoginItemsAE

Declared In

CFURL.h

CFURLHasDirectoryPath

Determines if a given URL's path represents a directory.

```
Boolean CFURLHasDirectoryPath (  
    CFURLRef anURL  
);
```

Parameters

anURL

The CFURL object to examine.

Return Value

true if *anURL* represents a directory, false otherwise.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

CFFTPSample

DisplayURL

ImageClient

MoreAppleEvents

Declared In

CFURL.h

CFURLResourceIsReachable

Returns whether the resource pointed to by a file URL can be reached.

```
Boolean CFURLResourceIsReachable (
    CFURLRef url,
    CFErrorRef *error
);
```

Parameters*url*

The URL to check.

error

The error that occurred in the case that the resource cannot be reached.

Return Value

true if the resource is reachable; otherwise, false.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLSetResourcePropertiesForKeys

Sets resource properties of a URL specified by a given dictionary of keys and values.

```
Boolean CFURLSetResourcePropertiesForKeys (
    CFURLRef url,
    CFDictionaryRef keyedPropertyValues,
    CFErrorRef *error
);
```

Parameters*url*

The URL.

keyedPropertyValues

A dictionary of resource values to be set.

error

The error that occurred in the case that one or more resource values cannot be set.

Return Valuetrue if all resource values in *keyedPropertyValues* are successfully set; otherwise, false.**Availability**

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLSetResourcePropertyForKey

Sets the resource property of the URL specified by a given key to a given value.

```
Boolean CFURLSetResourcePropertyForKey (
    CFURLRef url,
    CFStringRef key,
    CTypeRef propertyValue,
    CFErrorRef *error
);
```

Parameters

url

The URL.

key

The name of one of the URL's resource properties.

propertyValue

The value for the resource property defined by *key*.

error

The error that occurred in the case that the resource value cannot be set.

Return Value

true if the resource property named *key* is successfully set to *value*; otherwise, false.

Discussion

This function writes the new property value out to the backing store.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLSetTemporaryResourcePropertyForKey

Sets the resource property of the URL specified by a given key to a given value without writing the assignment to the backing store.

```
void CFURLSetTemporaryResourcePropertyForKey (
    CFURLRef url,
    CFStringRef key,
    CTypeRef propertyValue
);
```

Parameters

url

The URL.

key

The name of one of the URL's resource properties.

propertyValue

The value for the resource property defined by *key*.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLWriteBookmarkDataToFile

Creates an alias file on disk at a specified location with specified bookmark data.

```
Boolean CFURLWriteBookmarkDataToFile (
    CFDataRef bookmarkRef,
    CFURLRef fileURL,
    CFURLBookmarkFileCreationOptions options,
    CFErrorRef *errorRef
);
```

Parameters*bookmarkRef*

The bookmark data containing information for the alias file.

fileURL

The desired location of the alias file.

options

Options taken into account when creating the alias file.

errorRef

The error that occurred in the case that the alias file cannot be created.

Return Value

true if the alias file is successfully created; otherwise, false.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

Data Types

Bookmark Data Types

CFURLBookmarkCreationOptions

Type for bookmark data creation options.

```
typedef CFOptionFlags CFURLBookmarkCreationOptions;
```

Discussion

See [“Bookmark Data Creation Options”](#) (page 50) for possible values.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLBookmarkFileCreationOptions

Type for bookmark file creation options.

```
typedef CFOptionFlags CFURLBookmarkFileCreationOptions;
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

CFURLBookmarkResolutionOptions

Type for bookmark data resolution options.

```
typedef CFOptionFlags CFURLBookmarkResolutionOptions;
```

Discussion

See [“Bookmark Data Resolution Options”](#) (page 50) for possible values.

Availability

Available in Mac OS X v10.6 and later.

Declared In

CFURL.h

Miscellaneous

CFURLRef

A reference to a CFURL object.

```
typedef const struct __CFURL *CFURLRef;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFURL.h

Constants

Bookmark Data Constants

Bookmark Data Creation Options

Options used when creating bookmark data.

```
enum {
    kCFURLBookmarkCreationPreferFileIDResolutionMask = ( 1UL << 8 ),
    kCFURLBookmarkCreationMinimalBookmarkMask = ( 1UL << 9 ),
    kCFURLBookmarkCreationSuitableForBookmarkFile = ( 1UL << 10 )
};
```

Constants

`kCFURLBookmarkCreationPreferFileIDResolutionMask`

Option for specifying that an alias created with the bookmark data prefers resolving with its embedded file ID.

`kCFURLBookmarkCreationMinimalBookmarkMask`

Option for specifying that an alias created with the bookmark data be created with minimal information, which may make it smaller but still able to resolve in certain ways.

`kCFURLBookmarkCreationSuitableForBookmarkFile`

Option for specifying that the bookmark data include properties required to create Finder alias files.

Bookmark Data Resolution Options

Options used when resolving bookmark data.

```
enum {
    kCFBookmarkResolutionWithoutUIMask = ( 1UL << 8 ),
    kCFBookmarkResolutionWithoutMountingMask = ( 1UL << 9 ),
};
```

Constants

`kCFBookmarkResolutionWithoutUIMask`

Option for specifying that no UI feedback accompany resolution of the bookmark data.

`kCFBookmarkResolutionWithoutMountingMask`

Option for specifying that no volume should be mounted during resolution of the bookmark data.

File System Constants

Common File System Resource Keys

Keys that are applicable to file system URLs.

kCFURLNameKey
 kCFURLLocalizedNameKey
 kCFURLIsRegularFileKey
 kCFURLIsDirectoryKey
 kCFURLIsSymbolicLinkKey
 kCFURLIsVolumeKey
 kCFURLIsPackageKey
 kCFURLIsSystemImmutableKey
 kCFURLIsUserImmutableKey
 kCFURLIsHiddenKey
 kCFURLHasHiddenExtensionKey
 kCFURLCreationDateKey
 kCFURLContentAccessDateKey
 kCFURLContentModificationDateKey
 kCFURLAttributeModificationDateKey
 kCFURLLinkCountKey
 kCFURLParentDirectoryURLKey
 kCFURLVolumeURLKey
 kCFURLTypeIDentifierKey
 kCFURLLocalizedTypeDescriptionKey
 kCFURLLabelNumberKey
 kCFURLLabelColorKey
 kCFURLLocalizedLabelKey
 kCFURLEffectiveIconKey
 kCFURLCustomIconKey

Constants

kCFURLNameKey

Key for the resource's name in the file system, returned as a CFString object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLLocalizedNameKey

Key for the resource's localized or extension-hidden name, returned as a CFString object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLIsRegularFileKey

Key for determining whether the resource is a regular file, as opposed to a directory or a symbolic link. Returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLIsDirectoryKey

Key for determining whether the resource is a directory, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLIsSymbolicLinkKey

Key for determining whether the resource is a symbolic link, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

`kCFURLIsVolumeKey`

Key for determining whether the resource is the root directory of a volume, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLIsPackageKey`

Key for determining whether the resource is a packaged directory, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLIsSystemImmutableKey`

Key for determining whether the resource's system immutable bit is set, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLIsUserImmutableKey`

Key for determining whether the resource's user immutable bit is set, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLIsHiddenKey`

Key for determining whether the resource is normally not displayed to users, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLHasHiddenExtensionKey`

Key for determining whether the resource's extension is normally removed from its localized name, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLCreationDateKey`

Key for the resource's creation date, returned as a `CFDate` object if the volume supports creation dates, or `nil` if creation dates are unsupported.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLContentAccessDateKey`

Key for the last time the resource was accessed, returned as a `CFDate` object if the volume supports access dates, or `nil` if access dates are unsupported.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLContentModificationDateKey`

Key for the last time the resource was modified, returned as a `CFDate` object if the volume supports modification dates, or `nil` if modification dates are unsupported.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLAttributeModificationDateKey`

Key for the last time the resource's attributes were modified, returned as a `CFDate` object if the volume supports attribute modification dates, or `nil` if attribute modification dates are unsupported.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLLinkCountKey`

Key for the number of hard links to the resource, returned as a `CFNumber` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLParentDirectoryURLKey`

Key for the parent directory of the resource, returned as a `CFURL` object, or `nil` if the resource is the root directory of its volume.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLVolumeURLKey`

Key for the root directory of the resource's volume, returned as a `CFURL` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLTypeIDentifierKey`

Key for the resource's uniform type identifier (UTI), returned as a `CFString` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLLocalizedTypeDescriptionKey`

Key for the resource's localized type description, returned as a `CFString` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLLabelNumberKey`

Key for the resource's label number, returned as a `CFNumber` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLLabelColorKey`

Key for the resource's label color, returned as a `CGColorRef` object, or `nil` if the resource has no label color.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLLocalizedLabelKey`

Key for the resource's localized label text, returned as a `CFString` object, or `nil` if the resource has no localized label text.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLEffectiveIconKey`

Key for the resource's normal icon, returned as a `CGImageRef` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLCustomIconKey`

Key for the icon stored with the resource, returned as a `CGImageRef` object, or `nil` if the resource has no custom icon.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

File Property Keys

Keys that apply to properties of files.

`kCFURLFileSizeKey`

`kCFURLFileAllocatedSizeKey`

`kCFURLIsAliasFileKey`

Constants

`kCFURLFileSizeKey`

Key for the file's size in bytes, returned as a `CFNumber` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLFileAllocatedSizeKey`

Key for the total size allocated on disk for the file, returned as an `CFNumber` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLIsAliasFileKey`

Key for determining whether the file is an alias, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

Volume Property Keys

Keys that apply to volumes.

kCFURLVolumeLocalizedFormatDescriptionKey
 kCFURLVolumeTotalCapacityKey
 kCFURLVolumeAvailableCapacityKey
 kCFURLVolumeResourceCountKey
 kCFURLVolumeSupportsPersistentIDsKey
 kCFURLVolumeSupportsSymbolicLinksKey
 kCFURLVolumeSupportsHardLinksKey
 kCFURLVolumeSupportsJournalingKey
 kCFURLVolumeIsJournalingKey
 kCFURLVolumeSupportsSparseFilesKey
 kCFURLVolumeSupportsZeroRunsKey
 kCFURLVolumeSupportsCaseSensitiveNamesKey
 kCFURLVolumeSupportsCasePreservedNamesKey

Constants

kCFURLVolumeLocalizedFormatDescriptionKey

Key for the volume's descriptive format name, returned as a CFString object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeTotalCapacityKey

Key for the volume's capacity in bytes, returned as a CFNumber object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeAvailableCapacityKey

Key for the volume's available capacity in bytes, returned as a CFNumber object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeResourceCountKey

Key for the total number of resources on the volume, returned as a CFNumber object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeSupportsPersistentIDsKey

Key for determining whether the volume supports persistent IDs, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeSupportsSymbolicLinksKey

Key for determining whether the volume supports symbolic links, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeSupportsHardLinksKey

Key for determining whether the volume supports hard links, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

kCFURLVolumeSupportsJournalingKey

Key for determining whether the volume supports journaling, returned as a CFBoolean object.

Available in Mac OS X v10.6 and later.

Declared in CFURL.h.

`kCFURLVolumeIsJournalingKey`

Key for determining whether the volume is currently journaling, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLVolumeSupportsSparseFilesKey`

Key for determining whether the volume supports sparse files, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLVolumeSupportsZeroRunsKey`

Key for determining whether the volume supports zero runs, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLVolumeSupportsCaseSensitiveNamesKey`

Key for determining whether the volume supports case-sensitive names, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

`kCFURLVolumeSupportsCasePreservedNamesKey`

Key for determining whether the volume supports case-preserved names, returned as a `CFBoolean` object.

Available in Mac OS X v10.6 and later.

Declared in `CFURL.h`.

Miscellaneous

Component Type

The types of components in a URL.

```
typedef enum {
    kCFURLComponentScheme = 1,
    kCFURLComponentNetLocation = 2,
    kCFURLComponentPath = 3,
    kCFURLComponentResourceSpecifier = 4,
    kCFURLComponentUser = 5,
    kCFURLComponentPassword = 6,
    kCFURLComponentUserInfo = 7,
    kCFURLComponentHost = 8,
    kCFURLComponentPort = 9,
    kCFURLComponentParameterString = 10,
    kCFURLComponentQuery = 11,
    kCFURLComponentFragment = 12
} CFURLComponentType;
typedef enum CFURLPathStyle CFURLPathStyle;
```

Constants

`kCFURLComponentScheme`

The URL's scheme.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentNetLocation`

The URL's network location.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentPath`

The URL's path component.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentResourceSpecifier`

The URL's resource specifier.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentUser`

The URL's user.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentPassword`

The user's password.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentUserInfo`

The user's information.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentHost`

The URL's host.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentPort`

The URL's port.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentParameterString`

The URL's parameter string.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentQuery`

The URL's query.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

`kCFURLComponentFragment`

The URL's fragment.

Available in Mac OS X v10.3 and later.

Declared in `CFURL.h`.

Discussion

These constants are used by the [CFURLGetByteRangeForComponent](#) (page 41) function.

Availability

Available in Mac OS X v10.3 and later.

Path Style

Options you can use to determine how CFURL functions parse a file system path name.

```
enum CFURLPathStyle {
    kCFURLPOSIXPathStyle = 0,
    kCFURLHFSPathStyle = 1,
    kCFURLWindowsPathStyle = 2
};
typedef enum CFURLPathStyle CFURLPathStyle;
```

Constants

`kCFURLPOSIXPathStyle`

Indicates a POSIX style path name. Components are slash delimited. A leading slash indicates an absolute path; a trailing slash is not significant.

Available in Mac OS X v10.0 and later.

Declared in `CFURL.h`.

`kCFURLHFSPathStyle`

Indicates a HFS style path name. Components are colon delimited. A leading colon indicates a relative path, otherwise the first path component denotes the volume.

Available in Mac OS X v10.0 and later.

Declared in `CFURL.h`.

`kCFURLWindowsPathStyle`

Indicates a Windows style path name.

Available in Mac OS X v10.0 and later.

Declared in `CFURL.h`.

Document Revision History

This table describes the changes to *CFURL Reference*.

Date	Notes
2009-08-07	Added functions and constants for the bookmark data API new in Mac OS X v10.6.
2009-02-04	Corrected typos.
2008-07-11	Clarified the description of the CFURLGetBytes function.
2006-01-10	Clarified the behavior of the functions CFURLCreateStringByAddingPercentEscapes and CFURLGetFSRef.
2005-12-06	Made minor changes to clarify memory management rules.
2005-11-09	Removed reference to retired document.
2005-10-04	Corrected minor typographic errors.
2005-07-07	Clarified implementations of CFURLCanBeDecomposed and CFURLCreateCopyAppendingPathComponent, and description of CFURLCopyPathExtension.
2005-04-29	Moved Introduction to new Introduction page.
2004-08-31	Clarification of return values for CFURLCopyLastPathComponent.
2003-08-01	Added descriptions of new Mac OS X v10.3 API.
2003-01-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

B

Bookmark Data Creation Options [50](#)
Bookmark Data Resolution Options [50](#)

C

CFURLBookmarkCreationOptions **data type** [48](#)
CFURLBookmarkFileCreationOptions **data type** [49](#)
CFURLBookmarkResolutionOptions **data type** [49](#)
CFURLCanBeDecomposed **function** [11](#)
CFURLClearResourcePropertyCache **function** [11](#)
CFURLClearResourcePropertyCacheForKey **function** [12](#)
CFURLCopyAbsoluteURL **function** [12](#)
CFURLCopyFileSystemPath **function** [12](#)
CFURLCopyFragment **function** [13](#)
CFURLCopyHostName **function** [14](#)
CFURLCopyLastPathComponent **function** [14](#)
CFURLCopyNetLocation **function** [15](#)
CFURLCopyParameterString **function** [16](#)
CFURLCopyPassword **function** [16](#)
CFURLCopyPath **function** [17](#)
CFURLCopyPathExtension **function** [18](#)
CFURLCopyQueryString **function** [18](#)
CFURLCopyResourcePropertiesForKeys **function** [19](#)
CFURLCopyResourcePropertyForKey **function** [20](#)
CFURLCopyResourceSpecifier **function** [20](#)
CFURLCopyScheme **function** [21](#)
CFURLCopyStrictPath **function** [21](#)
CFURLCopyUserName **function** [22](#)
CFURLCreateAbsoluteURLWithBytes **function** [23](#)
CFURLCreateBookmarkData **function** [23](#)
CFURLCreateBookmarkDataFromAliasRecord **function** [24](#)
CFURLCreateBookmarkDataFromFile **function** [25](#)
CFURLCreateByResolvingBookmarkData **function** [25](#)
CFURLCreateCopyAppendingPathComponent **function** [26](#)
CFURLCreateCopyAppendingPathExtension **function** [27](#)
CFURLCreateCopyDeletingLastPathComponent **function** [28](#)
CFURLCreateCopyDeletingPathExtension **function** [28](#)
CFURLCreateData **function** [29](#)
CFURLCreateFilePathURL **function** [30](#)
CFURLCreateFileReferenceURL **function** [30](#)
CFURLCreateFromFileSystemRepresentation **function** [31](#)
CFURLCreateFromFileSystemRepresentationRelativeToBase **function** [31](#)
CFURLCreateFromFSRef **function** [32](#)
CFURLCreateResourcePropertiesForKeysFromBookmarkData **function** [33](#)
CFURLCreateResourcePropertyForKeyFromBookmarkData **function** [33](#)
CFURLCreateStringByAddingPercentEscapes **function** [34](#)
CFURLCreateStringByReplacingPercentEscapes **function** [35](#)
CFURLCreateStringByReplacingPercentEscapesUsingEncoding **function** [36](#)
CFURLCreateWithBytes **function** [37](#)
CFURLCreateWithFileSystemPath **function** [38](#)
CFURLCreateWithFileSystemPathRelativeToBase **function** [39](#)
CFURLCreateWithString **function** [39](#)
CFURLGetBaseURL **function** [40](#)
CFURLGetByteRangeForComponent **function** [41](#)
CFURLGetBytes **function** [41](#)
CFURLGetFileSystemRepresentation **function** [42](#)
CFURLGetFSRef **function** [43](#)
CFURLGetPortNumber **function** [44](#)
CFURLGetString **function** [44](#)
CFURLGetTypeID **function** [45](#)
CFURLHasDirectoryPath **function** [45](#)
CFURLRef **data type** [49](#)
CFURLResourceIsReachable **function** [46](#)
CFURLSetResourcePropertiesForKeys **function** [46](#)
CFURLSetResourcePropertyForKey **function** [47](#)

CFURLSetTemporaryResourcePropertyForKey
 function 47
 CFURLWriteBookmarkDataToFile function 48
 Common File System Resource Keys 50
 Component Type 56

F

File Property Keys 54

K

kCFBookmarkResolutionWithoutMountingMask
 constant 50
 kCFBookmarkResolutionWithoutUIMask constant 50
 kCFURLAttributeModificationDateKey constant 53
 kCFURLBookmarkCreationMinimalBookmarkMask
 constant 50
 kCFURLBookmarkCreationPreferFileIDResolutionMask
 constant 50
 kCFURLBookmarkCreationSuitableForBookmarkFile
 constant 50
 kCFURLComponentFragment constant 58
 kCFURLComponentHost constant 58
 kCFURLComponentNetLocation constant 57
 kCFURLComponentParameterString constant 58
 kCFURLComponentPassword constant 57
 kCFURLComponentPath constant 57
 kCFURLComponentPort constant 58
 kCFURLComponentQuery constant 58
 kCFURLComponentResourceSpecifier constant 57
 kCFURLComponentScheme constant 57
 kCFURLComponentUser constant 57
 kCFURLComponentUserInfo constant 57
 kCFURLContentAccessDateKey constant 52
 kCFURLContentModificationDateKey constant 52
 kCFURLCreationDateKey constant 52
 kCFURLCustomIconKey constant 54
 kCFURLEffectiveIconKey constant 54
 kCFURLFileAllocatedSizeKey constant 54
 kCFURLFileSizeKey constant 54
 kCFURLHasHiddenExtensionKey constant 52
 kCFURLHFSPathStyle constant 59
 kCFURLIsAliasFileKey constant 54
 kCFURLIsDirectoryKey constant 51
 kCFURLIsHiddenKey constant 52
 kCFURLIsPackageKey constant 52
 kCFURLIsRegularFileKey constant 51
 kCFURLIsSymbolicLinkKey constant 51
 kCFURLIsSystemImmutableKey constant 52

kCFURLIsUserImmutableKey constant 52
 kCFURLIsVolumeKey constant 52
 kCFURLLabelColorKey constant 53
 kCFURLLabelNumberKey constant 53
 kCFURLLinkCountKey constant 53
 kCFURLLocalizedLabelKey constant 53
 kCFURLLocalizedNameKey constant 51
 kCFURLLocalizedTypeDescriptionKey constant 53
 kCFURLNameKey constant 51
 kCFURLParentDirectoryURLKey constant 53
 kCFURLPOSIXPathStyle constant 58
 kCFURLTypeIDentifierKey constant 53
 kCFURLVolumeAvailableCapacityKey constant 55
 kCFURLVolumeIsJournalingKey constant 56
 kCFURLVolumeLocalizedFormatDescriptionKey
 constant 55
 kCFURLVolumeResourceCountKey constant 55
 kCFURLVolumeSupportsCasePreservedNamesKey
 constant 56
 kCFURLVolumeSupportsCaseSensitiveNamesKey
 constant 56
 kCFURLVolumeSupportsHardLinksKey constant 55
 kCFURLVolumeSupportsJournalingKey constant 55
 kCFURLVolumeSupportsPersistentIDsKey constant
 55
 kCFURLVolumeSupportsSparseFilesKey constant 56
 kCFURLVolumeSupportsSymbolicLinksKey constant
 55
 kCFURLVolumeSupportsZeroRunsKey constant 56
 kCFURLVolumeTotalCapacityKey constant 55
 kCFURLVolumeURLKey constant 53
 kCFURLWindowsPathStyle constant 59

P

Path Style 58

V

Volume Property Keys 54