
CFData Reference

Core Foundation



2006-02-07



Apple Inc.
© 2003, 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Carbon, and Cocoa 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

CFData Reference 5

- Overview 5
- Functions by Task 5
 - Creating a CFData Object 5
 - Examining a CFData Object 6
 - Getting the CFData Type ID 6
- Functions 6
 - CFDataGetTypeID 6
 - CFDataCreate 6
 - CFDataCreateCopy 7
 - CFDataCreateWithBytesNoCopy 8
 - CFDataGetBytePtr 9
 - CFDataGetBytes 10
 - CFDataGetLength 10
- Data Types 11
 - CFDataRef 11

Document Revision History 13

Index 15

CFData Reference

Derived From:	CFPropertyList : CType
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFData.h
Companion guides	Binary Data Programming Guide for Core Foundation Property List Programming Topics for Core Foundation

Overview

CFData and its derived mutable type, CFMutableData, provide support for data objects, object-oriented wrappers for byte buffers. Data objects let simple allocated buffers (that is, data with no embedded pointers) take on the behavior of Core Foundation objects. CFData creates static data objects, and CFMutableData creates dynamic data objects. Data objects are typically used for raw data storage.

You use the [CFDataCreate](#) (page 6) and [CFDataCreateCopy](#) (page 7) functions to create static data objects. These functions make a new copy of the supplied data. To create a data object that uses the supplied buffer instead of making a separate copy, use the [CFDataCreateWithBytesNoCopy](#) (page 8) function. You use the [CFDataGetBytes](#) (page 10) function to retrieve the bytes and the [CFDataGetLength](#) (page 10) function to get the length of the bytes.

CFData is “toll-free bridged” with its Cocoa Foundation counterpart, NSData. What this means is 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 `NSData *` parameter, you can pass in a `CFDataRef`, and in a function where you see a `CFDataRef` parameter, you can pass in an `NSData` instance. This also applies to concrete subclasses of `NSData`. See [Interchangeable Data Types](#) for more information on toll-free bridging.

Functions by Task

Creating a CFData Object

[CFDataCreate](#) (page 6)

Creates an immutable CFData object using data copied from a specified byte buffer.

[CFDataCreateCopy](#) (page 7)

Creates an immutable copy of a CFData object.

[CFDataCreateWithBytesNoCopy](#) (page 8)

Creates an immutable CFData object from an external (client-owned) byte buffer.

Examining a CFData Object

[CFDataGetBytePtr](#) (page 9)

Returns a read-only pointer to the bytes of a CFData object.

[CFDataGetBytes](#) (page 10)

Copies the byte contents of a CFData object to an external buffer.

[CFDataGetLength](#) (page 10)

Returns the number of bytes contained by a CFData object.

Getting the CFData Type ID

Functions

CFDataGetTypeID

Returns the type identifier for the CFData opaque type.

```
CTypeID CFDataGetTypeID (
    void
);
```

Return Value

The type identifier for the CFData opaque type.

Discussion

CFMutableData objects have the same type identifier as CFData objects.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

CFLocalServer

MoreSCF

RecentItems

Declared In

CFData.h

CFDataCreate

Creates an immutable CFData object using data copied from a specified byte buffer.

```
CFDataRef CFDataCreate (
    CFAllocatorRef allocator,
    const UInt8 *bytes,
    CFIndex length
);
```

Parameters*allocator*

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

bytes

A pointer to the byte buffer that contains the raw data to be copied into *theData*.

length

The number of bytes in the buffer (*bytes*).

Return Value

A new CFData object, or `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

Discussion

You must supply a count of the bytes in the buffer. This function always copies the bytes in the provided buffer into internal storage.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BackgroundExporter

BasicInputMethod

CFHostSample

DTSCarbonShell

TypeServicesForUnicode

Declared In

CFData.h

CFDataCreateCopy

Creates an immutable copy of a CFData object.

```
CFDataRef CFDataCreateCopy (
    CFAllocatorRef allocator,
    CFDataRef theData
);
```

Parameters*allocator*

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

theData

The CFData object to copy.

Return Value

An immutable copy of *theData*, or *NULL* if there was a problem creating the object. Ownership follows the Create Rule.

Discussion

The resulting object has the same byte contents as the original object, but it is always immutable. If the specified allocator and the allocator of the original object are the same, and the string is already immutable, this function may simply increment the retain count without making a true copy. To the caller, however, the resulting object is a true immutable copy, except the operation was more efficient.

Use this function when you need to pass a CFData object into another function by value (not reference).

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BackgroundExporter

Declared In

CFData.h

CFDataCreateWithBytesNoCopy

Creates an immutable CFData object from an external (client-owned) byte buffer.

```
CFDataRef CFDataCreateWithBytesNoCopy (
    CFAllocatorRef allocator,
    const UInt8 *bytes,
    CFIndex length,
    CFAllocatorRef bytesDeallocator
);
```

Parameters

allocator

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

bytes

A pointer to the byte buffer to be used as the backing store of the CFData object.

length

The number of bytes in the buffer *bytes*.

bytesDeallocator

The allocator to use to deallocate the external buffer when the CFData object is deallocated. If the default allocator is suitable for this purpose, pass *NULL* or *kCFAllocatorDefault*. If you do not want the created CFData object to deallocate the buffer (that is, you assume responsibility for freeing it yourself), pass *kCFAllocatorNull*.

Return Value

A new CFData object, or *NULL* if there was a problem creating the object. Ownership follows the Create Rule.

Discussion

This function creates an immutable CFData object from a buffer of unstructured bytes. Unless the situation warrants otherwise, the created object does not copy the external buffer to internal storage but instead uses the buffer as its backing store. However, you should never count on the object using the external buffer since it could copy the buffer to internal storage or might even dump the buffer altogether and use alternative means for storing the bytes.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

MoreIsBetter

QISA

Declared In

CFData.h

CFDataGetBytePtr

Returns a read-only pointer to the bytes of a CFData object.

```
const UInt8 * CFDataGetBytePtr (
    CFDataRef theData
);
```

Parameters

theData

The CFData object to examine.

Return Value

A read-only pointer to the bytes associated with *theData*.

Discussion

This function is guaranteed to return a pointer to a CFData object's internal bytes. CFData, unlike CFString, does not hide its internal storage.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BackgroundExporter

CFHostSample

CFLocalServer

MoreSCF

TypeServicesForUnicode

Declared In

CFData.h

CFDataGetBytes

Copies the byte contents of a CFData object to an external buffer.

```
void CFDataGetBytes (
    CFDataRef theData,
    CFRange range,
    UInt8 *buffer
);
```

Parameters

theData

The CFData object to examine.

range

The range of bytes in *theData* to get. To get all of the contents, pass `CFRangeMake(0,CFDataGetLength(theData))`.

buffer

A pointer to the byte buffer of length `range.length` that is allocated on the stack or heap. On return, the buffer contains the requested range of bytes.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BasicInputMethod

DTSCarbonShell

MoreIsBetter

QISA

QTCarbonShell

Declared In

CFData.h

CFDataGetLength

Returns the number of bytes contained by a CFData object.

```
CFIndex CFDataGetLength (
    CFDataRef theData
);
```

Parameters

theData

The CFData object to examine.

Return Value

An index that specifies the number of bytes in *theData*.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

CFLocalServer

MoreIsBetter

QISA

RecentItems

TypeServicesForUnicode

Declared In

CFData.h

Data Types

CFDataRef

A reference to an immutable CFData object.

```
typedef const struct __CFData *CFDataRef;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFData.h

Document Revision History

This table describes the changes to *CFData Reference*.

Date	Notes
2006-02-07	Corrected minor typographical error.
2005-12-06	Corrected links to companion documents and made minor changes to descriptive text for reference consistency.
2005-10-04	Clarified definition of type identifier.
2005-08-11	Cosmetic update to conform to document style guidelines.
2003-08-01	Added link to Carbon-Cocoa integration document.
2003-01-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

C

- CFDataCreate [function 6](#)
- CFDataCreateCopy [function 7](#)
- CFDataCreateWithBytesNoCopy [function 8](#)
- CFDataGetBytePtr [function 9](#)
- CFDataGetBytes [function 10](#)
- CFDataGetLength [function 10](#)
- CFDataGetTypeID [function 6](#)
- CFDataRef [data type 11](#)