
CFHTTPMessage Reference

Networking, Internet, & Web: Protocol Streams



2008-07-11



Apple Inc.
© 2008 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, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

CFHTTPMessage Reference 5

Overview	5
Functions by Task	5
Creating a Message	5
Modifying a message	5
Getting information from a message	6
Message authentication	6
Getting the CFHTTPMessage type identifier	6
Functions	7
CFHTTPMessageAddAuthentication	7
CFHTTPMessageAppendBytes	8
CFHTTPMessageApplyCredentialDictionary	8
CFHTTPMessageApplyCredentials	9
CFHTTPMessageCopyAllHeaderFields	10
CFHTTPMessageCopyBody	11
CFHTTPMessageCopyHeaderFieldValue	11
CFHTTPMessageCopyRequestMethod	12
CFHTTPMessageCopyRequestURL	12
CFHTTPMessageCopyResponseStatusLine	13
CFHTTPMessageCopySerializedMessage	13
CFHTTPMessageCopyVersion	14
CFHTTPMessageCreateCopy	14
CFHTTPMessageCreateEmpty	15
CFHTTPMessageCreateRequest	15
CFHTTPMessageCreateResponse	16
CFHTTPMessageGetResponseStatusCode	17
CFHTTPMessageGetTypeID	17
CFHTTPMessageIsHeaderComplete	18
CFHTTPMessageIsRequest	18
CFHTTPMessageSetBody	18
CFHTTPMessageSetHeaderFieldValue	19
Data Types	20
CFHTTPMessageRef	20
Constants	20
CFHTTP Version Constants	20
Authentication Schemes	20

Document Revision History 23

Index 25

CFHTTPMessage Reference

Derived From:	CType
Framework:	CoreServices
Declared in	CFHTTPAuthentication.h CFHTTPMessage.h
Companion guides	Getting Started with Networking CFNetwork Programming Guide

Overview

The `CFHTTPMessage` opaque type represents an HTTP message.

Functions by Task

Creating a Message

- [CFHTTPMessageCreateCopy](#) (page 14)
Gets a copy of a `CFHTTPMessage` object.
- [CFHTTPMessageCreateEmpty](#) (page 15)
Creates and returns a new, empty `CFHTTPMessage` object.
- [CFHTTPMessageCreateRequest](#) (page 15)
Creates and returns a `CFHTTPMessage` object for an HTTP request.
- [CFHTTPMessageCreateResponse](#) (page 16)
Creates and returns a `CFHTTPMessage` object for an HTTP response.

Modifying a message

- [CFHTTPMessageAppendBytes](#) (page 8)
Appends data to a `CFHTTPMessage` object.
- [CFHTTPMessageSetBody](#) (page 18)
Sets the body of a `CFHTTPMessage` object.
- [CFHTTPMessageSetHeaderFieldValue](#) (page 19)
Sets the value of a header field in an HTTP message.

Getting information from a message

[CFHTTPMessageCopyBody](#) (page 11)

Gets the body from a `CFHTTPMessage` object.

[CFHTTPMessageCopyAllHeaderFields](#) (page 10)

Gets all header fields from a `CFHTTPMessage` object.

[CFHTTPMessageCopyHeaderFieldValue](#) (page 11)

Gets the value of a header field from a `CFHTTPMessage` object.

[CFHTTPMessageCopyRequestMethod](#) (page 12)

Gets the request method from a `CFHTTPMessage` object.

[CFHTTPMessageCopyRequestURL](#) (page 12)

Gets the URL from a `CFHTTPMessage` object.

[CFHTTPMessageCopySerializedMessage](#) (page 13)

Serializes a `CFHTTPMessage` object.

[CFHTTPMessageCopyVersion](#) (page 14)

Gets the HTTP version from a `CFHTTPMessage` object.

[CFHTTPMessageIsRequest](#) (page 18)

Returns a boolean indicating whether the `CFHTTPMessage` is a request or a response.

[CFHTTPMessageIsHeaderComplete](#) (page 18)

Determines whether a message header is complete.

[CFHTTPMessageGetResponseStatusCode](#) (page 17)

Gets the status code from a `CFHTTPMessage` object representing an HTTP response.

[CFHTTPMessageCopyResponseStatusLine](#) (page 13)

Gets the status line from a `CFHTTPMessage` object.

Message authentication

[CFHTTPMessageApplyCredentials](#) (page 9)

Performs the authentication method specified by a `CFHTTPAuthentication` object.

[CFHTTPMessageApplyCredentialDictionary](#) (page 8)

Use a dictionary containing authentication credentials to perform the authentication method specified by a `CFHTTPAuthentication` object.

[CFHTTPMessageAddAuthentication](#) (page 7)

Adds authentication information to a request.

Getting the CFHTTPMessage type identifier

[CFHTTPMessageGetTypeID](#) (page 17)

Returns the Core Foundation type identifier for the `CFHTTPMessage` opaque type.

Functions

CFHTTPMessageAddAuthentication

Adds authentication information to a request.

```
Boolean CFHTTPMessageAddAuthentication (
    CFHTTPMessageRef request,
    CFHTTPMessageRef authenticationFailureResponse,
    CFStringRef username,
    CFStringRef password,
    CFStringRef authenticationScheme,
    Boolean forProxy
);
```

Parameters

request

The message to which to add authentication information.

authenticationFailureResponse

The response message that contains authentication failure information.

username

The username to add to the request.

password

The password to add to the request.

authenticationScheme

The authentication scheme to use (kCFHTTPAuthenticationSchemeBasic, kCFHTTPAuthenticationSchemeNegotiate, kCFHTTPAuthenticationSchemeNTLM, or kCFHTTPAuthenticationSchemeDigest), or pass NULL to use the strongest supported authentication scheme provided in the authenticationFailureResponse parameter.

forProxy

A flag indicating whether the authentication data that is being added is for a proxy's use (TRUE) or for a remote server's use (FALSE). If the error code provided by the authenticationFailureResponse parameter is 407, set forProxy to TRUE. If the error code is 401, set forProxy to FALSE.

Return Value

TRUE if the authentication information was successfully added, otherwise FALSE.

Discussion

This function adds the authentication information specified by the username, password, authenticationScheme, and forProxy parameters to the specified request message. The message referred to by the authenticationFailureResponse parameter typically contains a 401 or a 407 error code.

This function is best suited for sending a single request to the server. If you need to send multiple requests, use [CFHTTPMessageApplyCredentials](#) (page 9).

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageAppendBytes

Appends data to a `CFHTTPMessage` object.

```

Boolean CFHTTPMessageAppendBytes (
    CFHTTPMessageRef message,
    const UInt8 *newBytes,
    CFIndex numBytes
);

```

Parameters

message

The message to modify.

newBytes

A reference to the data to append.

numBytes

The length of the data pointed to by *newBytes*.

Return Value

TRUE if the data was successfully appended, otherwise FALSE.

Discussion

This function appends the data specified by *newBytes* to the specified message object which was created by calling [CFHTTPMessageCreateEmpty](#) (page 15). The data is an incoming serialized HTTP request or response received from a client or a server. While appending the data, this function deserializes it, removes any HTTP-based formatting that the message may contain, and stores the message in the message object. You can then call [CFHTTPMessageCopyVersion](#) (page 14), [CFHTTPMessageCopyBody](#) (page 11), [CFHTTPMessageCopyHeaderFieldValue](#) (page 11), and [CFHTTPMessageCopyAllHeaderFields](#) (page 10) to get the message's HTTP version, the message's body, a specific header field, and all of the message's headers, respectively.

If the message is a request, you can also call [CFHTTPMessageCopyRequestURL](#) (page 12) and [CFHTTPMessageCopyRequestMethod](#) (page 12) to get the message's request URL and request method, respectively.

If the message is a response, you can also call [CFHTTPMessageGetResponseStatusCode](#) (page 17) and [CFHTTPMessageCopyResponseStatusLine](#) (page 13) to get the message's status code and status line, respectively.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

`CFHTTPMessage.h`

CFHTTPMessageApplyCredentialDictionary

Use a dictionary containing authentication credentials to perform the authentication method specified by a `CFHTTPAuthentication` object.

```

Boolean CFHTTPMessageApplyCredentialDictionary (
    CFHTTPMessageRef request,
    CFHTTPAuthenticationRef auth,
    CFDictionaryRef dict,
    CFStreamError *error
);

```

Parameters*request*

The request for which the authentication method is to be performed.

auth

A `CFHTTPAuthentication` object specifying the authentication method to perform.

dict

A dictionary containing authentication credentials to be applied to the request. For information on the keys in this dictionary, see `CFHTTPAuthenticationRef`.

error

If an error occurs, upon return contains a `CFStreamError` object that describes the error and the error's domain. Pass `NULL` if you don't want to receive error information.

Return Value

`TRUE` if the authentication was successful, otherwise, `FALSE`.

Discussion

This function performs the authentication method specified by *auth* on behalf of the request specified by *request* using the credentials contained in the dictionary specified by *dict*. The dictionary must contain values for the `kCFHTTPAuthenticationUsername` and `kCFHTTPAuthenticationPassword` keys. If `CFHTTPAuthenticationRequiresAccountDomain` returns `TRUE` for *auth*, the dictionary must also contain a value for the `kCFHTTPAuthenticationAccountDomain` key.

Special Considerations

This function is thread safe as long as another thread does not alter the same `CFHTTPAuthentication` object at the same time.

Availability

Available in Mac OS X version 10.4 and later.

Related Sample Code

`ImageClient`

Declared In

`CFHTTPAuthentication.h`

CFHTTPMessageApplyCredentials

Performs the authentication method specified by a `CFHTTPAuthentication` object.

```
Boolean CFHTTPMessageApplyCredentials (
    CFHTTPMessageRef request,
    CFHTTPAuthenticationRef auth,
    CFStringRef username,
    CFStringRef password,
    CFStreamError *error
);
```

Parameters*request*

Request for which the authentication method is to be performed.

*auth*A `CFHTTPAuthentication` object specifying the authentication method to perform.*username*

Username for performing the authentication.

password

Password for performing the authentication.

*error*If an error occurs, upon return contains a `CFStreamError` object that describes the error and the error's domain. Pass `NULL` if you don't want to receive error information.**Return Value**

TRUE if the authentication was successful, otherwise, FALSE.

Discussion

This function performs the authentication method specified by `auth` on behalf of the request specified by `request` using the credentials specified by `username` and `password`. If, in addition to a username and password, you also need to specify an account domain, call [CFHTTPMessageApplyCredentialDictionary](#) (page 8) instead of this function.

This function is appropriate for performing several authentication requests. If you only need to make a single authentication request, consider using [CFHTTPMessageAddAuthentication](#) (page 7) instead.

Special Considerations

This function is thread safe as long as another thread does not alter the same `CFHTTPMessage` object at the same time.

Availability

Available in Mac OS X version 10.4 and later.

Declared In`CFHTTPAuthentication.h`**CFHTTPMessageCopyAllHeaderFields**Gets all header fields from a `CFHTTPMessage` object.

```
CFDictionaryRef CFHTTPMessageCopyAllHeaderFields (
    CFHTTPMessageRef message
);
```

Parameters*message*

The message to examine.

Return Value

A `CFDictionary` object containing keys and values that are `CFString` objects, where the key is the header fieldname and the dictionary value is the header field's value. Returns `NULL` if the header fields could not be copied. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCopyBodyGets the body from a `CFHTTPMessage` object.

```
CFDataRef CFHTTPMessageCopyBody (
    CFHTTPMessageRef message
);
```

Parameters*message*

The message to examine.

Return Value

A `CFData` object or `NULL` if there was a problem creating the object or if there is no message body. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCopyHeaderFieldValueGets the value of a header field from a `CFHTTPMessage` object.

```
CFStringRef CFHTTPMessageCopyHeaderFieldValue (
    CFHTTPMessageRef message,
    CFStringRef headerField
);
```

Parameters*message*

The message to examine.

headerField

The header field to copy.

Return Value

A `CFString` object containing a copy of the field specified by *headerField*, or `NULL` if there was a problem creating the object or if the specified header does not exist. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

`CFHTTPMessage.h`

CFHTTPMessageCopyRequestMethod

Gets the request method from a `CFHTTPMessage` object.

```
CFStringRef CFHTTPMessageCopyRequestMethod (
    CFHTTPMessageRef request
);
```

Parameters

request

The message to examine. This must be a request message.

Return Value

A `CFString` object containing a copy of the message's request method, or `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

`CFHTTPMessage.h`

CFHTTPMessageCopyRequestURL

Gets the URL from a `CFHTTPMessage` object.

```
CFURLRef CFHTTPMessageCopyRequestURL (
    CFHTTPMessageRef request
);
```

Parameters

request

The message to examine. This must be a request message.

Return Value

A `CFURLRef` object containing the URL or `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

ImageClient

Declared In

CFHTTPMessage.h

CFHTTPMessageCopyResponseStatusLine

Gets the status line from a CFHTTPMessage object.

```
CFStringRef CFHTTPMessageCopyResponseStatusLine (
    CFHTTPMessageRef response
);
```

Parameters*response*

The message to examine. This must be a response message.

Return Value

A string containing the message's status line, or NULL if there was a problem creating the object. The status line includes the message's protocol version and a success or error code. Ownership follows the Create Rule.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCopySerializedMessage

Serializes a CFHTTPMessage object.

```
CFDataRef CFHTTPMessageCopySerializedMessage (
    CFHTTPMessageRef request
);
```

Parameters*request*

The message to serialize.

Return Value

A CFData object containing the serialized message, or NULL if there was a problem creating the object. Ownership follows the Create Rule.

Discussion

This function returns a copy of a CFHTTPMessage object in serialized format that is ready for transmission.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCopyVersion

Gets the HTTP version from a CFHTTPMessage object.

```

CFStringRef CFHTTPMessageCopyVersion (
    CFHTTPMessageRef message
);

```

Parameters

message

The message to examine.

Return Value

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

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCreateCopy

Gets a copy of a CFHTTPMessage object.

```

CFHTTPMessageRef CFHTTPMessageCreateCopy (
    CFAllocatorRef alloc,
    CFHTTPMessageRef message
);

```

Parameters

allocator

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

message

The message to copy.

Return Value

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

Discussion

This function returns a copy of a CFHTTPMessage object that you can modify, for example, by calling [CFHTTPMessageCopyHeaderFieldValue](#) (page 11) or by calling [CFHTTPMessageSetBody](#) (page 18). Then serialize the message by calling [CFHTTPMessageCopySerializedMessage](#) (page 13) and send the serialized message to a client or a server.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageCreateEmpty

Creates and returns a new, empty `CFHTTPMessage` object.

```
CFHTTPMessageRef CFHTTPMessageCreateEmpty (
    CFAllocatorRef alloc,
    Boolean isRequest
);
```

Parameters

allocator

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

isRequest

A flag that determines whether to create an empty message request or an empty message response. Pass `TRUE` to create an empty request message; pass `FALSE` to create an empty response message.

Return Value

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

Discussion

Call [CFHTTPMessageAppendBytes](#) (page 8) to store an incoming, serialized HTTP request or response message in the empty message object.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

`CFHTTPMessage.h`

CFHTTPMessageCreateRequest

Creates and returns a `CFHTTPMessage` object for an HTTP request.

```
CFHTTPMessageRef CFHTTPMessageCreateRequest (
    CFAllocatorRef alloc,
    CFStringRef requestMethod,
    CFURLRef url,
    CFStringRef httpVersion
);
```

Parameters

allocator

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

requestMethod

The request method for the request. Use any of the request methods allowed by the HTTP version specified by *httpVersion*.

url

The URL to which the request will be sent.

httpVersion

The HTTP version for this message. Pass `kCFHTTPVersion1_0` or `kCFHTTPVersion1_1`.

Return Value

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

Discussion

This function returns a `CFHTTPMessage` object that you can use to build an HTTP request. Continue building the request by calling `CFHTTPMessageSetBody` (page 18) to set the message's body. Call `CFHTTPMessageCopyHeaderFieldValue` (page 11) to set the message's headers.

If you are using a `CFReadStream` object to send the message, call `CFReadStreamCreateForHTTPRequest` to create a read stream for the request. If you are not using `CFReadStream`, call `CFHTTPMessageCopySerializedMessage` (page 13) to make the message ready for transmission by serializing it.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

`CFNetworkHTTPDownload`
`ImageClient`

Declared In

`CFHTTPMessage.h`

CFHTTPMessageCreateResponse

Creates and returns a `CFHTTPMessage` object for an HTTP response.

```
CFHTTPMessageRef CFHTTPMessageCreateResponse (
    CFAllocatorRef allocator,
    CFIndex statusCode,
    CFStringRef statusDescription,
    CFStringRef httpVersion
);
```

Parameters

allocator

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

statusCode

The status code for this message response. The status code can be any of the status codes defined in section 6.1.1 of RFC 2616.

statusDescription

The description that corresponds to the status code. Pass `NULL` to use the standard description for the given status code, as found in RFC 2616.

httpVersion

The HTTP version for this message response. Pass `kCFHTTPVersion1_0` or `kCFHTTPVersion1_1`.

Return Value

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

Discussion

This function returns a `CFHTTPMessage` object that you can use to build an HTTP response. Continue building the response by calling `CFHTTPMessageSetBody` (page 18) to set the message's body. Call `CFHTTPMessageSetHeaderFieldValue` (page 19) to set the message's headers. Then call `CFHTTPMessageCopySerializedMessage` (page 13) to make the message ready for transmission by serializing it.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

`CFHTTPMessage.h`

CFHTTPMessageGetResponseStatusCode

Gets the status code from a `CFHTTPMessage` object representing an HTTP response.

```
CFIndex CFHTTPMessageGetResponseStatusCode (
    CFHTTPMessageRef response
);
```

Parameters

response

The message to examine. This must be a response message.

function result

The status code as defined by RFC 2616, section 6.1.1.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

ImageClient

Declared In

`CFHTTPMessage.h`

CFHTTPMessageGetTypeID

Returns the Core Foundation type identifier for the `CFHTTPMessage` opaque type.

```
CTypeID CFHTTPMessageGetTypeID ();
```

Return Value

The Core Foundation type identifier for the `CFHTTPMessage` opaque type.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageIsHeaderComplete

Determines whether a message header is complete.

```
Boolean CFHTTPMessageIsHeaderComplete (
    CFHTTPMessageRef message
);
```

Parameters

message

The message to verify.

function result

TRUE if the message header is complete, otherwise FALSE.

Discussion

After calling [CFHTTPMessageAppendBytes](#) (page 8), call this function to see if the message header is complete.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

CFHTTPMessage.h

CFHTTPMessageIsRequest

Returns a boolean indicating whether the CFHTTPMessage is a request or a response.

```
extern Boolean CFHTTPMessageIsRequest(CFHTTPMessageRef message);
```

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

CFHTTPMessageSetBody

Sets the body of a CFHTTPMessage object.

```
void CFHTTPMessageSetBody (
    CFHTTPMessageRef message,
    CFDataRef bodyData
);
```

Parameters

message

The message to modify.

bodyData

The data that is to be set as the body of the message.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

CFHTTPMessage.h

CFHTTPMessageSetHeaderFieldValue

Sets the value of a header field in an HTTP message.

```
void CFHTTPMessageSetHeaderFieldValue (
    CFHTTPMessageRef message,
    CFStringRef headerField,
    CFStringRef value
);
```

Parameters

message

The message to modify.

headerField

The header field to set.

value

The value to set.

Availability

Available in Mac OS X version 10.1 and later.

Related Sample Code

CocoaSOAP

Declared In

CFHTTPMessage.h

Data Types

CFHTTPMessageRef

An opaque reference representing an HTTP message.

```
typedef struct __CFHTTPMessage *CFHTTPMessageRef;
```

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CFHTTPMessage.h

Constants

CFHTTP Version Constants

Sets the HTTP version in a CFHTTPMessage request or response object.

```
const CFStringRef kCFHTTPVersion1_0;
const CFStringRef kCFHTTPVersion1_1;
```

Constants

kCFHTTPVersion1_0

Specifies HTTP version 1.0.

Available in Mac OS X version 10.1 and later.

kCFHTTPVersion1_1

Specifies HTTP version 1.1.

Available in Mac OS X version 10.1 and later.

Discussion

The HTTP version constants are used when you call [CFHTTPMessageCreateRequest](#) (page 15) and [CFHTTPMessageCreateResponse](#) (page 16) to create a request or response message.

Declared In

CFNetwork/CFHTTPMessage.h

Authentication Schemes

Constants used to specify the desired authentication scheme for a request.

```
extern const CFStringRef kCFHTTPAuthenticationSchemeBasic;  
extern const CFStringRef kCFHTTPAuthenticationSchemeDigest;  
extern const CFStringRef kCFHTTPAuthenticationSchemeNTLM;  
extern const CFStringRef kCFHTTPAuthenticationSchemeNegotiate;
```

Constants

`kCFHTTPAuthenticationSchemeBasic`

Request the HTTP basic authentication scheme.

Available in Mac OS X version 10.2 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeDigest`

Request the HTTP digest authentication scheme.

Available in Mac OS X version 10.2 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeNTLM`

Request the HTTP NTLM authentication scheme.

Available in Mac OS X version 10.5 and later.

Declared in `CFHTTPMessage.h`.

`kCFHTTPAuthenticationSchemeNegotiate`

Request an automatically negotiated authentication scheme.

Available in Mac OS X version 10.5 and later.

Declared in `CFHTTPMessage.h`.

Document Revision History

This table describes the changes to *CFHTTPMessage Reference*.

Date	Notes
2008-07-11	Added missing functions and constants.
2007-05-15	Made minor formatting changes.
2006-02-07	New document that describes the C API for creating, sending, receiving, and manipulating HTTP messages.

REVISION HISTORY

Document Revision History

Index

A

Authentication Schemes [20](#)

C

CFHTTP Version Constants [20](#)

CFHTTPMessageAddAuthentication [function 6, 7](#)

CFHTTPMessageAppendBytes [function 5, 8](#)

CFHTTPMessageApplyCredentialDictionary [function 8](#)

CFHTTPMessageApplyCredentials [function 9](#)

CFHTTPMessageCopyAllHeaderFields [function 6, 10](#)

CFHTTPMessageCopyBody [function 6, 11](#)

CFHTTPMessageCopyHeaderFieldValue [function 6, 11](#)

CFHTTPMessageCopyRequestMethod [function 12](#)

CFHTTPMessageCopyRequestURL [function 6, 12](#)

CFHTTPMessageCopyResponseStatusLine [function 13](#)

CFHTTPMessageCopySerializedMessage [function 6, 13](#)

CFHTTPMessageCopyVersion [function 6, 14](#)

CFHTTPMessageCreateCopy [function 5, 14](#)

CFHTTPMessageCreateEmpty [function 5, 15](#)

CFHTTPMessageCreateRequest [function 5, 15](#)

CFHTTPMessageCreateResponse [function 5, 16](#)

CFHTTPMessageCreateResponse [function 16](#)

CFHTTPMessageGetResponseStatusCode [function 6, 13, 17](#)

CFHTTPMessageGetTypeID [function 6, 17](#)

CFHTTPMessageHeaderComplete [function 6, 18](#)

CFHTTPMessageIsHeaderComplete [function 18](#)

CFHTTPMessageIsRequest [function 18](#)

CFHTTPMessageRef [structure 20](#)

CFHTTPMessageSetBody [function 5, 18](#)

CFHTTPMessageSetHeaderFieldValue [function 5, 19](#)

F

functions

CFHTTPMessageAddAuthentication [6, 7](#)

CFHTTPMessageAppendBytes [5, 8](#)

CFHTTPMessageCopyAllHeaderFields [6, 10](#)

CFHTTPMessageCopyBody [6, 11](#)

CFHTTPMessageCopyHeaderFieldValue [6, 11](#)

CFHTTPMessageCopyRequestURL [6, 12](#)

CFHTTPMessageCopySerializedMessage [6, 13](#)

CFHTTPMessageCopyVersion [6, 14](#)

CFHTTPMessageCreateCopy [5, 14](#)

CFHTTPMessageCreateEmpty [5, 15](#)

CFHTTPMessageCreateRequest [5, 15](#)

CFHTTPMessageCreateResponse [5, 16](#)

CFHTTPMessageGetResponseStatusCode [6, 13, 17](#)

CFHTTPMessageGetTypeID [6, 17](#)

CFHTTPMessageHeaderComplete [6, 18](#)

CFHTTPMessageSetBody [5, 18](#)

CFHTTPMessageSetHeaderFieldValues [5, 19](#)

K

kCFHTTPAuthenticationSchemeBasic [constant 21](#)

kCFHTTPAuthenticationSchemeDigest [constant 21](#)

kCFHTTPAuthenticationSchemeNegotiate [constant 21](#)

kCFHTTPAuthenticationSchemeNTLM [constant 21](#)

kCFHTTPVersion1_0 [constant 20](#)

kCFHTTPVersion1_1 [constant 20](#)