
NSURLHandle Class Reference

[Cocoa > Networking](#)



2009-02-04



Apple Inc.
© 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, Cocoa, 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

NSURLHandle Class Reference 5

- Overview 5
- Tasks 5
 - Constructing NSURLHandles 5
 - Managing Subclasses 6
 - Managing Clients 6
 - Setting and Getting Resource Properties 6
 - Loading Resource Data 6
 - Writing Resource Data 7
- Constants 7
 - NSURLHandleStatus 7

Appendix A Deprecated NSURLHandle Methods 9

- Deprecated in Mac OS X v10.4 and later 9
 - cachedHandleForURL: 9
 - canInitWithURL: 9
 - registerURLHandleClass: 10
 - URLHandleClassForURL: 10
 - addClient: 11
 - availableResourceData 11
 - backgroundLoadDidFailWithReason: 11
 - beginLoadInBackground 12
 - cancelLoadInBackground 12
 - didLoadBytes:loadComplete: 12
 - endLoadInBackground 13
 - expectedResourceDataSize 13
 - failureReason 14
 - flushCachedData 14
 - initWithURL:cached: 14
 - loadInBackground 15
 - loadInForeground 15
 - propertyForKey: 15
 - propertyForKeyIfAvailable: 16
 - removeClient: 16
 - resourceData 17
 - status 17
 - writeData: 17
 - writeProperty:forKey: 18

Document Revision History 19

Index 21

NSURLHandle Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Declared in	NSURLHandle.h

`NSURLHandle` is deprecated in Mac OS X v10.4 and later. Applications that are intended for deployment on Mac OS X v10.3 or later should use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.

Overview

`NSURLHandle` declares the programmatic interface for an object that accesses and manages resource data indicated by an `NSURL` object. A single `NSURLHandle` can service multiple equivalent `NSURL` objects, but only if these URLs map to the same resource.

Cocoa provides private concrete subclasses to handle HTTP and file URL schemes. If you want to implement support for additional URL schemes, you would do so by creating a subclass of `NSURLHandle`. You can use `NSURL` and `NSURLHandle` to download from FTP sites without subclassing.

Tasks

Constructing NSURLHandles

- + `cachedHandleForURL:` (page 9) **Deprecated in Mac OS X v10.4 and later**
Returns the URL handle from the cache that has serviced the specified URL or another identical URL. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `initWithURL:cached:` (page 14) **Deprecated in Mac OS X v10.4 and later**
Initializes a newly created URL handle with the specified URL. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Managing Subclasses

- + `canInitWithURL:` (page 9) **Deprecated in Mac OS X v10.4 and later**
Returns whether a URL handle can be initialized with a given URL. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- + `registerURLHandleClass:` (page 10) **Deprecated in Mac OS X v10.4 and later**
Registers a subclass of `NSURLHandle` as an available subclass for handling URLs (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- + `URLHandleClassForURL:` (page 10) **Deprecated in Mac OS X v10.4 and later**
Returns the class of the URL handle that will be used for a specified URL. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Managing Clients

- `addClient:` (page 11) **Deprecated in Mac OS X v10.4 and later**
Adds a client of the URL handle. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `removeClient:` (page 16) **Deprecated in Mac OS X v10.4 and later**
Removes `client` as an `NSURLHandleClient` of the receiver. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Setting and Getting Resource Properties

- `propertyForKey:` (page 15) **Deprecated in Mac OS X v10.4 and later**
Returns the property for the specified key. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `propertyForKeyIfAvailable:` (page 16) **Deprecated in Mac OS X v10.4 and later**
Returns the property for the specified key only if the value is already available; that is, the client doesn't need to do any work. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `writeProperty:forKey:` (page 18) **Deprecated in Mac OS X v10.4 and later**
Sets the property of the receiver's resource for a specified key to the specified value. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Loading Resource Data

- `availableResourceData` (page 11) **Deprecated in Mac OS X v10.4 and later**
Immediately returns the currently available resource data managed by the URL handle. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `backgroundLoadDidFailWithReason:` (page 11) **Deprecated in Mac OS X v10.4 and later**
Called when a background load fails. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- `beginLoadInBackground` (page 12) **Deprecated in Mac OS X v10.4 and later**
Called when a background load begins. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- [cancelLoadInBackground](#) (page 12) **Deprecated in Mac OS X v10.4 and later**
Called to cancel a load currently in progress. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [didLoadBytes:loadComplete:](#) (page 12) **Deprecated in Mac OS X v10.4 and later**
Appends new data to the receiver's resource data. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [endLoadInBackground](#) (page 13) **Deprecated in Mac OS X v10.4 and later**
Halts any background loading. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [expectedResourceDataSize](#) (page 13) **Deprecated in Mac OS X v10.4 and later**
Returns the expected length of the resource data if it is provided by the server. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [failureReason](#) (page 14) **Deprecated in Mac OS X v10.4 and later**
Returns a string describing the reason a load failed. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [flushCachedData](#) (page 14) **Deprecated in Mac OS X v10.4 and later**
Flushes any cached data for the URL served by this URL handle. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [loadInBackground](#) (page 15) **Deprecated in Mac OS X v10.4 and later**
Loads the receiver's data in the background. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [loadInForeground](#) (page 15) **Deprecated in Mac OS X v10.4 and later**
Loads the receiver's data synchronously. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [resourceData](#) (page 17) **Deprecated in Mac OS X v10.4 and later**
Returns the resource data managed by the receiver, loading it if necessary. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)
- [status](#) (page 17) **Deprecated in Mac OS X v10.4 and later**
Returns the status of the receiver. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Writing Resource Data

- [writeData:](#) (page 17) **Deprecated in Mac OS X v10.4 and later**
Attempts to write a specified set of data to the location specified by the receiver's URL. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Constants

NSURLHandleStatus

These following constants are defined by `NSURLHandle` and are returned by [status](#) (page 17).

```
typedef enum {  
    NSURLHandleNotLoaded = 0,  
    NSURLHandleLoadSucceeded,  
    NSURLHandleLoadInProgress,  
    NSURLHandleLoadFailed  
} NSURLHandleStatus;
```

Constants

NSURLHandleNotLoaded

The resource data has not been loaded. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Available in Mac OS X v10.0 and later.

Declared in `NSURLHandle.h`.

NSURLHandleLoadSucceeded

The resource data was successfully loaded. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Available in Mac OS X v10.0 and later.

Declared in `NSURLHandle.h`.

NSURLHandleLoadInProgress

The resource data is in the process of loading. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Available in Mac OS X v10.0 and later.

Declared in `NSURLHandle.h`.

NSURLHandleLoadFailed

The resource data failed to load. (**Deprecated.** Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

Available in Mac OS X v10.0 and later.

Declared in `NSURLHandle.h`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSURLHandle.h`

Deprecated NSURLHandle Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.4 and later

cachedHandleForURL:

Returns the URL handle from the cache that has serviced the specified URL or another identical URL. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
+ (NSURLHandle *)cachedHandleForURL:(NSURL *)aURL
```

Parameters

aURL

The URL whose cached URL handle is desired.

Return Value

The URL handle from the cache that has serviced *aURL* or another identical URL. Returns `nil` if there is no such handle.

Discussion

Subclasses of `NSURLHandle` must override this method.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

canInitWithURL:

Returns whether a URL handle can be initialized with a given URL. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
+ (BOOL)canInitWithURL:(NSURL *)aURL
```

Parameters

aURL

The URL in question.

Return Value

YES if a URL handle can be initialized with *aURL*, NO otherwise.

Deprecated NSURLHandle Methods

Discussion

Subclasses of `NSURLHandle` must override this method to identify which URLs they can service.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

registerURLHandleClass:

Registers a subclass of `NSURLHandle` as an available subclass for handling URLs (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
+ (void)registerURLHandleClass:(Class)aURLHandleSubclass
```

Parameters

aURLHandleSubclass

The new subclass to register as an available subclass.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

URLHandleClassForURL:

Returns the class of the URL handle that will be used for a specified URL. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
+ (Class)URLHandleClassForURL:(NSURL *)aURL
```

Parameters

aURL

The URL in question.

Return Value

The class of the URL handle that will be used for *aURL*.

Discussion

Subclasses of `NSURLHandle` must be registered via the `registerURLHandleClass:` (page 10) method. The subclass is determined by asking the list of registered subclasses if it `canInitWithURL:` (page 9); the first class to respond YES is selected.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

addClient:

Adds a client of the URL handle. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (void)addClient:(id < NSURLHandleClient >)client
```

Parameters

client

An object conforming to the `NSURLHandleClient` protocol.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

availableResourceData

Immediately returns the currently available resource data managed by the URL handle. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (NSData *)availableResourceData
```

Return Value

The currently available resource data managed by the URL handle. Returns `nil` if a previous attempt to load the data failed.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

backgroundLoadDidFailWithReason:

Called when a background load fails. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (void)backgroundLoadDidFailWithReason:(NSString *)reason
```

Parameters

reason

The status message indicating why the background load failed.

Discussion

This method is provided mainly for subclasses that wish to take some action before passing along the failure notification to the URL client. This method should invoke `super`'s implementation before returning.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

`NSURLHandle.h`

beginLoadInBackground

Called when a background load begins. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)beginLoadInBackground

Discussion

This method is provided mainly for subclasses that wish to take advantage of the superclass failure-reporting mechanism.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [endLoadInBackground](#) (page 13)
- [loadInBackground](#) (page 15)

Declared In

NSURLHandle.h

cancelLoadInBackground

Called to cancel a load currently in progress. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)cancelLoadInBackground

Discussion

This method is provided mainly for subclasses that wish to take some action before a background load is canceled. This method should invoke `super`'s implementation before returning.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [endLoadInBackground](#) (page 13)

Declared In

NSURLHandle.h

didLoadBytes:loadComplete:

Appends new data to the receiver's resource data. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)didLoadBytes:(NSData *)newBytes loadComplete:(BOOL)done

Parameters

newBytes

The newly loaded bytes.

Deprecated NSURLHandle Methods

done

YES if `newBytes` contains the last piece of data for the URL, NO otherwise.

Discussion

You should call this method when loading the resource data in the background.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [loadInBackground](#) (page 15)

Declared In

NSURLHandle.h

endLoadInBackground

Halts any background loading. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)endLoadInBackground

Discussion

This method is called by [cancelLoadInBackground](#) (page 12).

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [beginLoadInBackground](#) (page 12)
- [cancelLoadInBackground](#) (page 12)
- [loadInBackground](#) (page 15)

Declared In

NSURLHandle.h

expectedResourceDataSize

Returns the expected length of the resource data if it is provided by the server. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (long long)expectedResourceDataSize

Return Value

The expected size of the resource data, in bytes. A negative value if the length is unknown.

Discussion

This information can be queried before all the data has arrived.

Availability

Deprecated in Mac OS X v10.4 and later.

Available in Mac OS X v10.3 and later.

Deprecated NSURLHandle Methods

Declared In

NSURLHandle.h

failureReason

Returns a string describing the reason a load failed. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (NSString *)failureReason

Return Value

A string describing the reason a load failed. If the load has not failed, returns `nil`.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

flushCachedData

Flushes any cached data for the URL served by this URL handle. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)flushCachedData

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

initWithURL:cached:

Initializes a newly created URL handle with the specified URL. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (id)initWithURL:(NSURL *)aURL cached:(BOOL)willCache

Parameters

aURL

The URL for the new handle.

willCache

YES if the URL handle should cache its data and respond to requests from equivalent URLs for the cached data, NO otherwise.

Discussion

Subclasses of `NSURLHandle` must override this method.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

loadInBackground

Loads the receiver's data in the background. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (void)loadInBackground

Discussion

Each subclass determines its own loading policy. Clients should not assume that multiple background loads can proceed simultaneously. For example, a subclass may maintain only one thread for background loading, so only one background loading operation can be in progress at a time. If multiple background loads are requested, the later requests will wait in a queue until earlier requests are handled.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [beginLoadInBackground](#) (page 12)

Declared In

NSURLHandle.h

loadInForeground

Loads the receiver's data synchronously. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (NSData *)loadInForeground

Return Value

The loaded data.

Discussion

Called by [resourceData](#) (page 17). Subclasses of `NSURLHandle` must override this method.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

propertyForKey:

Returns the property for the specified key. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (id)propertyForKey:(NSString *)propertyKey

Deprecated NSURLHandle Methods

Parameters*propertyKey*

The key of the desired property.

Return ValueThe value associated with *propertyKey*. Returns *nil* if there is no such key.**Discussion**

Subclasses of NSURLHandle must override this method.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [writeProperty:forKey:](#) (page 18)
- [propertyForKeyIfAvailable:](#) (page 16)

Declared In

NSURLHandle.h

propertyForKeyIfAvailable:

Returns the property for the specified key only if the value is already available; that is, the client doesn't need to do any work. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (id)propertyForKeyIfAvailable:(NSString *)propertyKey
```

Parameters*propertyKey*

The key of the desired property.

Return ValueThe value associated with *propertyKey*. Returns *nil* if there is no such key or if the client would have to do work to fetch the property.**Availability**

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

removeClient:

Removes *client* as an `NSURLHandleClient` of the receiver. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (void)removeClient:(id < NSURLHandleClient >)client
```

Parameters*client*An object conforming to the `NSURLHandleClient` protocol.

Deprecated NSURLHandle Methods

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

resourceData

Returns the resource data managed by the receiver, loading it if necessary. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (NSData *)resourceData

Return Value

The resource data managed by the receiver.

Discussion

Blocks until all data is available.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

status

Returns the status of the receiver. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (NSURLHandleStatus)status

Return Value

The status of the receiver. Possible return statuses are described in “Constants” (page 7).

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

writeData:

Attempts to write a specified set of data to the location specified by the receiver’s URL. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

- (BOOL)writeData:(NSData *)data

Parameters

data

The data to write.

Deprecated NSURLHandle Methods

Return Value

YES if successful, NO otherwise.

Discussion

Must be overridden by subclasses.

Availability

Deprecated in Mac OS X v10.4 and later.

Declared In

NSURLHandle.h

writeProperty:forKey:

Sets the property of the receiver's resource for a specified key to the specified value. (Deprecated in Mac OS X v10.4 and later. Use `NSURLConnection` or `NSURLDownload` instead; see *URL Loading System*.)

```
- (BOOL)writeProperty:(id)propertyValue forKey:(NSString *)propertyKey
```

Parameters

propertyValue

The new value for the property.

propertyKey

The key of the desired property.

Return Value

YES if the modification was successful, NO otherwise.

Discussion

Must be overridden by subclasses.

Availability

Deprecated in Mac OS X v10.4 and later.

See Also

- [propertyForKey:](#) (page 15)

Declared In

NSURLHandle.h

Document Revision History

This table describes the changes to *NSURLHandle Class Reference*.

Date	Notes
2009-02-04	Corrected typos.
2006-06-28	Updated the Constants section to conform to reference consistency guidelines.
2006-05-23	Updated to reflect that this class is deprecated in Mac OS X v10.4 and later.
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`addClient`: [instance method 11](#)
`availableResourceData` [instance method 11](#)

B

`backgroundLoadDidFailWithReason`: [instance method 11](#)
`beginLoadInBackground` [instance method 12](#)

C

`cachedHandleForURL`: [class method 9](#)
`cancelLoadInBackground` [instance method 12](#)
`canInitWithURL`: [class method 9](#)

D

`didLoadBytes:loadComplete`: [instance method 12](#)

E

`endLoadInBackground` [instance method 13](#)
`expectedResourceDataSize` [instance method 13](#)

F

`failureReason` [instance method 14](#)
`flushCachedData` [instance method 14](#)

I

`initWithURL:cached`: [instance method 14](#)

L

`loadInBackground` [instance method 15](#)
`loadInForeground` [instance method 15](#)

N

`NSURLHandleLoadFailed` [constant 8](#)
`NSURLHandleLoadInProgress` [constant 8](#)
`NSURLHandleLoadSucceeded` [constant 8](#)
`NSURLHandleNotLoaded` [constant 8](#)
`NSURLHandleStatus` [data type 7](#)

P

`propertyForKey`: [instance method 15](#)
`propertyForKeyIfAvailable`: [instance method 16](#)

R

`registerURLHandleClass`: [class method 10](#)
`removeClient`: [instance method 16](#)
`resourceData` [instance method 17](#)

S

`status` [instance method 17](#)

U

`URLHandleClassForURL:` **class method** [10](#)

W

`writeData:` **instance method** [17](#)

`writeProperty:forKey:` **instance method** [18](#)