

---

# NSURLResponse Class Reference

Networking, Internet, & Web: Protocol Streams





Apple Inc.  
© 2007 Apple Inc.  
All rights reserved.

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

The Apple logo is a trademark of Apple Inc.

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

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

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

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

Apple and the Apple logo 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

---

## **NSURLResponse Class Reference 5**

---

Overview	5
Adopted Protocols	5
Tasks	6
Creating a Response	6
Getting the Response Properties	6
Instance Methods	6
expectedContentLength	6
initWithURL:MIMEType:expectedContentLength:textEncodingName:	7
MIMEType	7
suggestedFilename	8
textEncodingName	8
URL	9
Constants	9
Response Length Unknown Error	9

---

## **Document Revision History 11**

---

## **Index 13**

---



# NSURLResponse Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.2 with Safari 1.0 installed. Available in Mac OS X v10.2.7 and later.
<b>Companion guide</b>	URL Loading System
<b>Declared in</b>	NSURLResponse.h
<b>Related sample code</b>	CocoaSOAP LSMSmartCategorizer QuickLookDownloader URL CacheInfo XMLBrowser

## Overview

`NSURLResponse` declares the programmatic interface for an object that accesses the response returned by an `NSURLRequest` instance.

`NSURLResponse` encapsulates the metadata associated with a URL load in a manner independent of protocol and URL scheme.

`NSHTTPURLResponse` is a subclass of `NSURLResponse` that provides methods for accessing information specific to HTTP protocol responses. An `NSHTTPURLResponse` object represents a response to an HTTP URL load request.

**Note:** `NSURLResponse` objects do not contain the actual bytes representing the content of a URL. See `NSURLConnection` for more information about receiving the content data for a URL load.

## Adopted Protocols

NSCoding

```

initWithCoder:
initWithCoder:

```

**NSCopying**

```
copyWithZone:
```

## Tasks

### Creating a Response

- [initWithURL:MIMETYPE:expectedContentLength:textEncodingName:](#) (page 7)  
Returns an initialized `NSURLResponse` object with the URL, MIME type, length, and text encoding set to given values.

### Getting the Response Properties

- [expectedContentLength](#) (page 6)  
Returns the receiver's expected content length
- [suggestedFilename](#) (page 8)  
Returns a suggested filename for the response data.
- [MIMETYPE](#) (page 7)  
Returns the receiver's MIME type.
- [textEncodingName](#) (page 8)  
Returns the name of the receiver's text encoding provided by the response's originating source.
- [URL](#) (page 9)  
Returns the receiver's URL.

## Instance Methods

### **expectedContentLength**

Returns the receiver's expected content length

```
-(long long)expectedContentLength
```

**Return Value**

The receiver's expected content length, or `NSURLResponseUnknownLength` if the length can't be determined.

**Discussion**

Some protocol implementations report the content length as part of the response, but not all protocols guarantee to deliver that amount of data. Clients should be prepared to deal with more or less data.

**Availability**

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

### Related Sample Code

QuickLookDownloader

### Declared In

NSURLResponse.h

## initWithURL:MIMEType:expectedContentLength:textEncodingName:

Returns an initialized `NSURLResponse` object with the URL, MIME type, length, and text encoding set to given values.

```
- (id)initWithURL:(NSURL *)URL MIMEType:(NSString *)MIMEType
    expectedContentLength:(NSInteger)length textEncodingName:(NSString *)name
```

### Parameters

*URL*

The URL for the new object.

*MIMEType*

The MIME type.

*length*

The expected content length. This value should be -1 if the expected length is undetermined

*name*

The text encoding name. This value may be `nil`.

### Return Value

An initialized `NSURLResponse` object with the URL set to *URL*, the MIME type set to *MIMEType*, length set to *length*, and text encoding name set to *name*.

### Discussion

This is the designated initializer for `NSURLResponse`.

### Availability

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

### Declared In

NSURLResponse.h

## MIMEType

Returns the receiver's MIME type.

```
- (NSString *)MIMEType
```

### Return Value

The receiver's MIME type.

**Discussion**

The MIME type is often provided by the response's originating source. However, that value may be changed or corrected by a protocol implementation if it can be determined that the response's source reported the information incorrectly.

If the response's originating source does not provide a MIME type, an attempt to guess the MIME type may be made.

**Availability**

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

**Declared In**

NSURLResponse.h

**suggestedFilename**

Returns a suggested filename for the response data.

- (NSString \*)suggestedFilename

**Return Value**

A suggested filename for the response data.

**Discussion**

The method tries to create a filename using the following, in order:

1. A filename specified using the content disposition header.
2. The last path component of the URL.
3. The host of the URL.

If the host of URL can't be converted to a valid filename, the filename "unknown" is used.

In most cases, this method appends the proper file extension based on the MIME type. This method will always return a valid filename regardless of whether or not the resource is saved to disk.

**Availability**

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

**Declared In**

NSURLResponse.h

**textEncodingName**

Returns the name of the receiver's text encoding provided by the response's originating source.

- (NSString \*)textEncodingName

**Return Value**

The name of the receiver's text encoding provided by the response's originating source, or `nil` if no text encoding was provided by the protocol

**Discussion**

Clients can convert this string to an `NSStringEncoding` or a `CFStringEncoding` using the methods and functions available in the appropriate framework.

**Availability**

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

**Related Sample Code**

XMLBrowser

**Declared In**

`NSURLResponse.h`

**URL**

Returns the receiver's URL.

```
- (NSURL *)URL
```

**Return Value**

The receiver's URL.

**Availability**

Available in Mac OS X v10.2 with Safari 1.0 installed.

Available in Mac OS X v10.2.7 and later.

**Declared In**

`NSURLResponse.h`

## Constants

**Response Length Unknown Error**

The following error code is returned by [expectedContentLength](#) (page 6).

```
#define NSURLResponseUnknownLength ((long long)-1)
```

**Constants**

`NSURLResponseUnknownLength`

Returned when the response length cannot be determined in advance of receiving the data from the server. For example, `NSURLResponseUnknownLength` is returned when the server HTTP response does not include a Content-Length header.

Available in Mac OS X v10.2 and later.

Declared in `NSURLResponse.h`.



# Document Revision History

---

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

Date	Notes
2007-12-11	Added documentation for the <code>NSURLResponseUnknownLength</code> macro.
2006-05-23	First publication of this content as a separate document.
	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## E

---

`expectedContentLength` [instance method 6](#)

## I

---

`initWithURL:MIMEType:expectedContentLength:  
textEncodingName:` [instance method 7](#)

## M

---

`MIMEType` [instance method 7](#)

## N

---

`NSURLResponseUnknownLength` [constant 9](#)

## R

---

`Response Length Unknown Error` [9](#)

## S

---

`suggestedFilename` [instance method 8](#)

## T

---

`textEncodingName` [instance method 8](#)

## U

---

`URL` [instance method 9](#)