

---

# NSXMLParser Class Reference

[Cocoa](#) > [Data Management](#)



2007-02-02



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, 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

## **NSXMLParser Class Reference 5**

---

Overview	5
Tasks	5
Initializing a Parser Object	5
Managing Delegates	6
Managing Parser Behavior	6
Parsing	6
Handling XML	6
Handling the DTD	7
Obtaining Parser State	8
Instance Methods	8
abortParsing	8
columnNumber	8
delegate	9
initWithContentsOfURL:	9
initWithData:	10
lineNumber	10
parse	10
parserError	11
publicID	11
setDelegate:	11
setShouldProcessNamespaces:	12
setShouldReportNamespacePrefixes:	12
setShouldResolveExternalEntities:	13
shouldProcessNamespaces	13
shouldReportNamespacePrefixes	14
shouldResolveExternalEntities	14
systemID	15
Delegate Methods	15
parser:didEndElement:namespaceURI:qualifiedName:	15
parser:didEndMappingPrefix:	16
parser:didStartElement:namespaceURI:qualifiedName:attributes:	16
parser:didStartMappingPrefix:toURI:	17
parser:foundAttributeDeclarationWithName:forElement:type:defaultValue:	17
parser:foundCDATA:	18
parser:foundCharacters:	18
parser:foundComment:	19
parser:foundElementDeclarationWithName:model:	19
parser:foundExternalEntityDeclarationWithName:publicID:systemID:	19
parser:foundIgnorableWhitespace:	20
parser:foundInternalEntityDeclarationWithName:value:	21

- parser:foundNotationDeclarationWithName:publicID:systemID: 21
- parser:foundProcessingInstructionWithTarget:data: 22
- parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName: 22
- parser:parseErrorOccurred: 23
- parser:resolveExternalEntityName:systemID: 23
- parser:validationErrorOccurred: 24
- parserDidEndDocument: 24
- parserDidStartDocument: 25
- Constants 25
  - NSXMLParserErrorDomain 25
  - NSXMLParserError 25
  - Parser Error Constants 26

---

**Document Revision History 39**

---

**Index 41**

---

# NSXMLParser Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.3 and later.
<b>Companion guide</b>	Event-Driven XML Programming Guide for Cocoa
<b>Declared in</b>	NSXMLParser.h
<b>Related sample code</b>	ImageMapExample

## Overview

Instances of this class parse XML documents (including DTD declarations) in an event-driven manner. An `NSXMLParser` notifies its delegate about the items (elements, attributes, CDATA blocks, comments, and so on) that it encounters as it processes an XML document. It does not itself do anything with those parsed items except report them. It also reports parsing errors. For convenience, an `NSXMLParser` object in the following descriptions is sometimes referred to as a parser object.

**Note:** Namespace support was implemented in `NSXMLParser` for Mac OS X v10.4. Namespace-related methods of `NSXMLParser` prior to this version have no effect.

## Tasks

### Initializing a Parser Object

- [initWithContentsOfURL:](#) (page 9)  
Initializes the receiver with the XML content referenced by the given URL.
- [initWithData:](#) (page 10)  
Initializes the receiver with the XML contents encapsulated in a given data object.

## Managing Delegates

- [setDelegate:](#) (page 11)  
Sets the receiver's delegate.
- [delegate](#) (page 9)  
Returns the receiver's delegate.

## Managing Parser Behavior

- [setShouldProcessNamespaces:](#) (page 12)  
Specifies whether the receiver reports the namespace and the qualified name of an element in related delegation methods .
- [shouldProcessNamespaces](#) (page 13)  
Indicates whether the receiver reports the namespace and the qualified name of an element in related delegation methods.
- [setShouldReportNamespacePrefixes:](#) (page 12)  
Specifies whether the receiver reports the scope of namespace declarations using related delegation methods.
- [shouldReportNamespacePrefixes](#) (page 14)  
Indicates whether the receiver reports the prefixes indicating the scope of namespace declarations using related delegation methods.
- [setShouldResolveExternalEntities:](#) (page 13)  
Specifies whether the receiver reports declarations of external entities using the delegate method [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19).
- [shouldResolveExternalEntities](#) (page 14)  
Indicates whether the receiver reports declarations of external entities using the delegate method [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19).

## Parsing

- [parse](#) (page 10)  
Starts the event-driven parsing operation.
- [abortParsing](#) (page 8)  
Stops the parser object.
- [parserError](#) (page 11)  
Returns an `NSError` object from which you can obtain information about a parsing error.

## Handling XML

- [parserDidStartDocument:](#) (page 25) *delegate method*  
Sent by the parser object to the delegate when it begins parsing a document.
- [parserDidEndDocument:](#) (page 24) *delegate method*  
Sent by the parser object to the delegate when it has successfully completed parsing.

- `parser:didStartElement:namespaceURI:qualifiedName:attributes:` (page 16) *delegate method*  
Sent by a parser object to its delegate when it encounters a start tag for a given element.
- `parser:didEndElement:namespaceURI:qualifiedName:` (page 15) *delegate method*  
Sent by a parser object to its delegate when it encounters an end tag for a specific element.
- `parser:didStartMappingPrefix:toURI:` (page 17) *delegate method*  
Sent by a parser object to its delegate the first time it encounters a given namespace prefix, which is mapped to a URI.
- `parser:didEndMappingPrefix:` (page 16) *delegate method*  
Sent by a parser object to its delegate when a given namespace prefix goes out of scope.
- `parser:resolveExternalEntityName:systemID:` (page 23) *delegate method*  
Sent by a parser object to its delegate when it encounters a given external entity with a specific system ID.
- `parser:parseErrorOccurred:` (page 23) *delegate method*  
Sent by a parser object to its delegate when it encounters a fatal error.
- `parser:validationErrorOccurred:` (page 24) *delegate method*  
Sent by a parser object to its delegate when it encounters a fatal validation error. NSXMLParser currently does not invoke this method and does not perform validation.
- `parser:foundCharacters:` (page 18) *delegate method*  
Sent by a parser object to provide its delegate with a string representing all or part of the characters of the current element.
- `parser:foundIgnorableWhitespace:` (page 20) *delegate method*  
Reported by a parser object to provide its delegate with a string representing all or part of the ignorable whitespace characters of the current element.
- `parser:foundProcessingInstructionWithTarget:data:` (page 22) *delegate method*  
Sent by a parser object to its delegate when it encounters a processing instruction.
- `parser:foundComment:` (page 19) *delegate method*  
Sent by a parser object to its delegate when it encounters a comment in the XML.
- `parser:foundCDATA:` (page 18) *delegate method*  
Sent by a parser object to its delegate when it encounters a CDATA block.

## Handling the DTD

- `parser:foundAttributeDeclarationWithName:forElement:type:defaultValue:` (page 17) *delegate method*  
Sent by a parser object to its delegate when it encounters a declaration of an attribute that is associated with a specific element.
- `parser:foundElementDeclarationWithName:model:` (page 19) *delegate method*  
Sent by a parser object to its delegate when it encounters a declaration of an element with a given model.
- `parser:foundExternalEntityDeclarationWithName:publicID:systemID:` (page 19) *delegate method*  
Sent by a parser object to its delegate when it encounters an external entity declaration.
- `parser:foundInternalEntityDeclarationWithName:value:` (page 21) *delegate method*  
Sent by a parser object to the delegate when it encounters an internal entity declaration.

- [parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName:](#) (page 22) *delegate method*  
Sent by a parser object to its delegate when it encounters an unparsed entity declaration.
- [parser:foundNotationDeclarationWithName:publicID:systemID:](#) (page 21) *delegate method*  
Sent by a parser object to its delegate when it encounters a notation declaration.

## Obtaining Parser State

- [columnNumber](#) (page 8)  
Returns the column number of the XML document being processed by the receiver.
- [lineNumber](#) (page 10)  
Returns the line number of the XML document being processed by the receiver.
- [publicID](#) (page 11)  
Returns the public identifier of the external entity referenced in the XML document.
- [systemID](#) (page 15)  
Returns the system identifier of the external entity referenced in the XML document.

## Instance Methods

### abortParsing

Stops the parser object.

- (void)abortParsing

#### Discussion

If you invoke this method, the delegate, if it implements [parser:parseErrorOccurred:](#) (page 23), is informed of the cancelled parsing operation.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [parse](#) (page 10)
- [parserError](#) (page 11)

#### Related Sample Code

ImageMapExample

#### Declared In

NSXMLParser.h

### columnNumber

Returns the column number of the XML document being processed by the receiver.

- (NSInteger)columnNumber



**Discussion**

The column refers to the nesting level of the XML elements in the document. You may invoke this method once a parsing operation has begun or after an error occurs.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [lineNumber](#) (page 10)

**Declared In**

NSXMLParser.h

**delegate**

Returns the receiver's delegate.

- (id)delegate

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setDelegate:](#) (page 11)

**Declared In**

NSXMLParser.h

**initWithContentsOfURL:**

Initializes the receiver with the XML content referenced by the given URL.

- (id)initWithContentsOfURL:(NSURL \*)url

**Parameters**

*url*

An NSURL object specifying a URL. The URL must be fully qualified and refer to a scheme that is supported by the NSURL class.

**Return Value**

An initialized NSXMLParser object or nil if an error occurs.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [initWithData:](#) (page 10)

**Declared In**

NSXMLParser.h

## initWithData:

Initializes the receiver with the XML contents encapsulated in a given data object.

```
- (id)initWithData:(NSData *)data
```

### Parameters

*data*

An NSData object containing XML markup.

### Return Value

An initialized NSXMLParser object or nil if an error occurs.

### Discussion

This method is the designated initializer.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [initWithContentsOfURL:](#) (page 9)

### Declared In

NSXMLParser.h

## lineNumber

Returns the line number of the XML document being processed by the receiver.

```
- (NSInteger)lineNumber
```

### Discussion

You may invoke this method once a parsing operation has begun or after an error occurs.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [columnNumber](#) (page 8)

### Declared In

NSXMLParser.h

## parse

Starts the event-driven parsing operation.

```
- (BOOL)parse
```

### Return Value

YES if parsing is successful and NO in there is an error or if the parsing operation is aborted.

### Availability

Available in Mac OS X v10.3 and later.

**See Also**

- [abortParsing](#) (page 8)
- [parserError](#) (page 11)

**Related Sample Code**

ImageMapExample

**Declared In**

NSXMLParser.h

## parserError

Returns an NSError object from which you can obtain information about a parsing error.

- (NSError \*)parserError

**Discussion**

You may invoke this method after a parsing operation abnormally terminates to determine the cause of error.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [abortParsing](#) (page 8)
- [parse](#) (page 10)

**Declared In**

NSXMLParser.h

## publicID

Returns the public identifier of the external entity referenced in the XML document.

- (NSString \*)publicID

**Discussion**

You may invoke this method once a parsing operation has begun or after an error occurs.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [systemID](#) (page 15)

**Declared In**

NSXMLParser.h

## setDelegate:

Sets the receiver's delegate.

- (void)setDelegate:(id)delegate

**Parameters***delegate*

An object that is the new delegate. It is not retained.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [delegate](#) (page 9)

**Related Sample Code**

ImageMapExample

**Declared In**

NSXMLParser.h

**setShouldProcessNamespaces:**

Specifies whether the receiver reports the namespace and the qualified name of an element in related delegation methods .

- (void)setShouldProcessNamespaces:(BOOL)shouldProcessNamespaces

**Parameters***shouldProcessNamespaces*

YES if the receiver should report the namespace and qualified name of each element, NO otherwise.  
The default value is NO.

**Discussion**

The invoked delegation methods are

[parser:didStartElement:namespaceURI:qualifiedName:attributes:](#) (page 16) and  
[parser:didEndElement:namespaceURI:qualifiedName:](#) (page 15).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [shouldProcessNamespaces](#) (page 13)

**Declared In**

NSXMLParser.h

**setShouldReportNamespacePrefixes:**

Specifies whether the receiver reports the scope of namespace declarations using related delegation methods.

- (void)setShouldReportNamespacePrefixes:(BOOL)shouldReportNamespacePrefixes

**Parameters***shouldReportNamespacePrefixes*

YES if the receiver should report the scope of namespace declarations, NO otherwise. The default value is NO.

**Discussion**

The invoked delegation methods are [parser:didStartMappingPrefix:toURI:](#) (page 17) and [parser:didEndMappingPrefix:](#) (page 16).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [shouldReportNamespacePrefixes](#) (page 14)

**Declared In**

NSXMLParser.h

**setShouldResolveExternalEntities:**

Specifies whether the receiver reports declarations of external entities using the delegate method [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19).

- (void)setShouldResolveExternalEntities:(BOOL)shouldResolveExternalEntities

**Parameters**

*shouldResolveExternalEntities*

YES if the receiver should report declarations of external entities, NO otherwise. The default value is NO.

**Discussion**

If you pass in YES, you may cause other I/O operations, either network-based or disk-based, to load the external DTD.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [shouldResolveExternalEntities](#) (page 14)

**Declared In**

NSXMLParser.h

**shouldProcessNamespaces**

Indicates whether the receiver reports the namespace and the qualified name of an element in related delegation methods.

- (BOOL)shouldProcessNamespaces

**Return Value**

YES if the receiver reports namespace and qualified name, NO otherwise.

**Discussion**

The invoked delegation methods are [parser:didStartElement:namespaceURI:qualifiedName:attributes:](#) (page 16) and [parser:didEndElement:namespaceURI:qualifiedName:](#) (page 15).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setShouldProcessNamespaces:](#) (page 12)

**Declared In**

NSXMLParser.h

**shouldReportNamespacePrefixes**

Indicates whether the receiver reports the prefixes indicating the scope of namespace declarations using related delegation methods.

- (BOOL)shouldReportNamespacePrefixes

**Return Value**

YES if the receiver reports the scope of namespace declarations, NO otherwise. The default value is NO.

**Discussion**

The invoked delegation methods are [parser:didStartMappingPrefix:toURI:](#) (page 17) and [parser:didEndMappingPrefix:](#) (page 16).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setShouldReportNamespacePrefixes:](#) (page 12)

**Declared In**

NSXMLParser.h

**shouldResolveExternalEntities**

Indicates whether the receiver reports declarations of external entities using the delegate method [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19).

- (BOOL)shouldResolveExternalEntities

**Return Value**

YES if the receiver reports declarations of external entities, NO otherwise. The default value is NO.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setShouldResolveExternalEntities:](#) (page 13)

**Declared In**

NSXMLParser.h

## systemID

Returns the system identifier of the external entity referenced in the XML document.

- (NSString \*)systemID

### Discussion

You may invoke this method once a parsing operation has begun or after an error occurs.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [publicID](#) (page 11)

### Declared In

NSXMLParser.h

## Delegate Methods

### parser:didEndElement:namespaceURI:qualifiedName:

Sent by a parser object to its delegate when it encounters an end tag for a specific element.

- (void)parser:(NSXMLParser \*)parser didEndElement:(NSString \*)elementName  
namespaceURI:(NSString \*)namespaceURI qualifiedName:(NSString \*)qName

### Parameters

*parser*

A parser object.

*elementName*

A string that is the name of an element (in its end tag).

*namespaceURI*

If namespace processing is turned on, contains the URI for the current namespace as a string object.

*qName*

If namespace processing is turned on, contains the qualified name for the current namespace as a string object..

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [parser:didStartElement:namespaceURI:qualifiedName:attributes:](#) (page 16)

- [setShouldProcessNamespaces:](#) (page 12)

### Declared In

NSXMLParser.h

**parser:didEndMappingPrefix:**

Sent by a parser object to its delegate when a given namespace prefix goes out of scope.

```
- (void)parser:(NSXMLParser *)parser didEndMappingPrefix:(NSString *)prefix
```

**Parameters**

*parser*

A parser object.

*prefix*

A string that is a namespace prefix.

**Discussion**

The parser sends this message only when namespace-prefix reporting is turned on through the [setShouldReportNamespacePrefixes:](#) (page 12) method.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:didStartMappingPrefix:toURI:](#) (page 17)

**Declared In**

NSXMLParser.h

**parser:didStartElement:namespaceURI:qualifiedName:attributes:**

Sent by a parser object to its delegate when it encounters a start tag for a given element.

```
- (void)parser:(NSXMLParser *)parser didStartElement:(NSString *)elementName
    namespaceURI:(NSString *)namespaceURI qualifiedName:(NSString *)qualifiedName
    attributes:(NSDictionary *)attributeDict
```

**Parameters**

*parser*

A parser object.

*elementName*

A string that is the name of an element (in its start tag).

*namespaceURI*

If namespace processing is turned on, contains the URI for the current namespace as a string object.

*qualifiedName*

If namespace processing is turned on, contains the qualified name for the current namespace as a string object..

*attributeDict*

A dictionary that contains any attributes associated with the element. Keys are the names of attributes, and values are attribute values.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:didEndElement:namespaceURI:qualifiedName:](#) (page 15)

- [setShouldProcessNamespaces:](#) (page 12)



**Declared In**

NSXMLParser.h

**parser:didStartMappingPrefix:toURI:**

Sent by a parser object to its delegate the first time it encounters a given namespace prefix, which is mapped to a URI.

```
- (void)parser:(NSXMLParser *)parser didStartMappingPrefix:(NSString *)prefix
  toURI:(NSString *)namespaceURI
```

**Parameters***parser*

A parser object.

*prefix*

A string that is a namespace prefix.

*namespaceURI*

A string that specifies a namespace URI.

**Discussion**

The parser object sends this message only when namespace-prefix reporting is turned on through the [setShouldReportNamespacePrefixes:](#) (page 12) method.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:didEndMappingPrefix:](#) (page 16)

**Declared In**

NSXMLParser.h

**parser:foundAttributeDeclarationWithName:forElement:type:defaultValue:**

Sent by a parser object to its delegate when it encounters a declaration of an attribute that is associated with a specific element.

```
- (void)parser:(NSXMLParser *)parser foundAttributeDeclarationWithName:(NSString *)attributeName
  forElement:(NSString *)elementName type:(NSString *)type
  defaultValue:(NSString *)defaultValue
```

**Parameters***parser*

An NSXMLParser object parsing XML.

*attributeName*

A string that is the name of an attribute.

*elementName*A string that is the name of an element that has the attribute *attributeName*.*type*

A string, such as "ENTITY", "NOTATION", or "ID", that indicates the type of the attribute.

*defaultValue*

A string that specifies the default value of the attribute.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [parser:didStartElement:namespaceURI:qualifiedName:attributes:](#) (page 16)

#### Declared In

NSXMLParser.h

### parser:foundCDATA:

Sent by a parser object to its delegate when it encounters a CDATA block.

```
- (void)parser:(NSXMLParser *)parser foundCDATA:(NSData *)CDATABlock
```

#### Parameters

*parser*

An NSXMLParser object parsing XML.

*CDATABlock*

A data object containing a block of CDATA.

#### Discussion

Through this method the parser object passes the contents of the block to its delegate in an NSData object. The CDATA block is character data that is ignored by the parser. The encoding of the character data is UTF-8. To convert the data object to a string object, use the NSString method `initWithData:encoding:`.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

NSXMLParser.h

### parser:foundCharacters:

Sent by a parser object to provide its delegate with a string representing all or part of the characters of the current element.

```
- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string
```

#### Parameters

*parser*

A parser object.

*string*

A string representing the complete or partial textual content of the current element.

#### Discussion

The parser object may send the delegate several `parser:foundCharacters:` messages to report the characters of an element. Because *string* may be only part of the total character content for the current element, you should append it to the current accumulation of characters until the element changes.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSXMLParser.h

**parser:foundComment:**

Sent by a parser object to its delegate when it encounters a comment in the XML.

```
- (void)parser:(NSXMLParser *)parser foundComment:(NSString *)comment
```

**Parameters**

*parser*

An NSXMLParser object parsing XML.

*comment*

A string that is the content of a comment in the XML.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSXMLParser.h

**parser:foundElementDeclarationWithName:model:**

Sent by a parser object to its delegate when it encounters a declaration of an element with a given model.

```
- (void)parser:(NSXMLParser *)parser foundElementDeclarationWithName:(NSString *)elementName model:(NSString *)model
```

**Parameters**

*parser*

An NSXMLParser object parsing XML.

*elementName*

A string that is the name of an element.

*model*

A string that specifies a model for *elementName*.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:didStartElement:namespaceURI:qualifiedName:attributes:](#) (page 16)

**Declared In**

NSXMLParser.h

**parser:foundExternalEntityDeclarationWithName:publicID:systemID:**

Sent by a parser object to its delegate when it encounters an external entity declaration.

```
- (void)parser:(NSXMLParser *)parser foundExternalEntityDeclarationWithName:(NSString *)entityName publicID:(NSString *)publicID systemID:(NSString *)systemID
```

**Parameters***parser*

An NSXMLParser object parsing XML.

*entityName*

A string that is the name of an entity.

*publicID*A string that specifies the public ID associated with *entityName*.*systemID*A string that specifies the system ID associated with *entityName*.**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:foundInternalEntityDeclarationWithName:value:](#) (page 21)
- [parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName:](#) (page 22)
- [parser:resolveExternalEntityName:systemID:](#) (page 23)

**Declared In**

NSXMLParser.h

**parser:foundIgnorableWhitespace:**

Reported by a parser object to provide its delegate with a string representing all or part of the ignorable whitespace characters of the current element.

```
- (void)parser:(NSXMLParser *)parser foundIgnorableWhitespace:(NSString *)whitespaceString
```

**Parameters***parser*

A parser object.

*whitespaceString*

A string representing all or part of the ignorable whitespace characters of the current element.

**Discussion**

All the whitespace characters of the element (including carriage returns, tabs, and new-line characters) may not be provided through an individual invocation of this method. The parser may send the delegate several `parser:foundIgnorableWhitespace:` messages to report the whitespace characters of an element. You should append the characters in each invocation to the current accumulation of characters.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:foundCharacters:](#) (page 18)

**Declared In**

NSXMLParser.h

**parser:foundInternalEntityDeclarationWithName:value:**

Sent by a parser object to the delegate when it encounters an internal entity declaration.

```
- (void)parser:(NSXMLParser *)parser foundInternalEntityDeclarationWithName:(NSString *)name value:(NSString *)value
```

**Parameters***parser*

An NSXMLParser object parsing XML.

*name*

A string that is the declared name of an internal entity.

*value*A string that is the value of entity *name*.**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19)
- [parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName:](#) (page 22)

**Declared In**

NSXMLParser.h

**parser:foundNotationDeclarationWithName:publicID:systemID:**

Sent by a parser object to its delegate when it encounters a notation declaration.

```
- (void)parser:(NSXMLParser *)parser foundNotationDeclarationWithName:(NSString *)name publicID:(NSString *)publicID systemID:(NSString *)systemID
```

**Parameters***parser*

An NSXMLParser object parsing XML.

*name*

A string that is the name of the notation.

*publicID*A string specifying the public ID associated with the notation *name*.*systemID*A string specifying the system ID associated with the notation *name*.**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSXMLParser.h

**parser:foundProcessingInstructionWithTarget:data:**

Sent by a parser object to its delegate when it encounters a processing instruction.

```
- (void)parser:(NSXMLParser *)parser foundProcessingInstructionWithTarget:(NSString *)target data:(NSString *)data
```

**Parameters**

*parser*

A parser object.

*target*

A string representing the target of a processing instruction.

*data*

A string representing the data for a processing instruction.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSXMLParser.h

**parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName:**

Sent by a parser object to its delegate when it encounters an unparsed entity declaration.

```
- (void)parser:(NSXMLParser *)parser foundUnparsedEntityDeclarationWithName:(NSString *)name publicID:(NSString *)publicID systemID:(NSString *)systemID notationName:(NSString *)notationName
```

**Parameters**

*parser*

An NSXMLParser object parsing XML.

*name*

A string that is the name of the unparsed entity in the declaration.

*publicID*

A string specifying the public ID associated with the entity *name*.

*systemID*

A string specifying the system ID associated with the entity *name*.

*notationName*

A string specifying a notation of the declaration of entity *name*.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19)
- [parser:foundInternalEntityDeclarationWithName:value:](#) (page 21)
- [parser:resolveExternalEntityName:systemID:](#) (page 23)

**Declared In**

NSXMLParser.h

**parser:parseErrorOccurred:**

Sent by a parser object to its delegate when it encounters a fatal error.

```
- (void)parser:(NSXMLParser *)parser parseErrorOccurred:(NSError *)parseError
```

**Parameters**

*parser*

A parser object.

*parseError*

An NSError object describing the parsing error that occurred.

**Discussion**

When this method is invoked, parsing is stopped. For further information about the error, you can query *parseError* or you can send the receiver a [parserError](#) (page 11) message. You can also send the parser [lineNumber](#) (page 10) and [columnNumber](#) (page 8) messages to further isolate where the error occurred. Typically you implement this method to display information about the error to the user.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:validationErrorOccurred:](#) (page 24)

**Declared In**

NSXMLParser.h

**parser:resolveExternalEntityName:systemID:**

Sent by a parser object to its delegate when it encounters a given external entity with a specific system ID.

```
- (NSData *)parser:(NSXMLParser *)parser resolveExternalEntityName:(NSString *)entityName systemID:(NSString *)systemID
```

**Parameters**

*parser*

A parser object.

*entityName*

A string that specifies the external name of an entity.

*systemID*

A string that specifies the system ID for the external entity.

**Return Value**

An NSData object that contains the resolution of the given external entity.

**Discussion**

The delegate can resolve the external entity (for example, locating and reading an externally declared DTD) and provide the result to the parser object as an NSData object.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:foundExternalEntityDeclarationWithName:publicID:systemID:](#) (page 19)

- [parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notationName:](#) (page 22)

**Declared In**

NSXMLParser.h

**parser:validationErrorOccurred:**

Sent by a parser object to its delegate when it encounters a fatal validation error. NSXMLParser currently does not invoke this method and does not perform validation.

- (void)parser:(NSXMLParser \*)*parser* validationErrorOccurred:(NSError \*)*validError*

**Parameters**

*parser*

A parser object.

*validError*

An NSError object describing the validation error that occurred.

**Discussion**

If you want to validate an XML document, use the validation features of the NSXMLDocument class.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parser:parseErrorOccurred:](#) (page 23)

**Declared In**

NSXMLParser.h

**parserDidEndDocument:**

Sent by the parser object to the delegate when it has successfully completed parsing.

- (void)parserDidEndDocument:(NSXMLParser \*)*parser*

**Parameters**

*parser*

A parser object.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parserDidStartDocument:](#) (page 25)

**Declared In**

NSXMLParser.h



**parserDidStartDocument:**

Sent by the parser object to the delegate when it begins parsing a document.

```
- (void)parserDidStartDocument:(NSXMLParser *)parser
```

**Parameters**

*parser*

A parser object.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [parserDidEndDocument:](#) (page 24)

**Declared In**

NSXMLParser.h

## Constants

**NSXMLParserErrorDomain**

This constant defines the NSXMLParser error domain.

```
NSString * const NSXMLParserErrorDomain
```

**Constants**

NSXMLParserErrorDomain

Indicates an error in XML parsing.

Used by NSError.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

**Declared In**

NSXMLParser.h

**NSXMLParserError**

A type defined for the constants listed in [“Parser Error Constants”](#) (page 26).

```
typedef NSInteger NSXMLParserError;
```

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSXMLParser.h

## Parser Error Constants

The following error types are defined by `NSXMLParser`.

```

typedef enum {
    NSXMLParserInternalError = 1,
    NSXMLParserOutOfMemoryError = 2,
    NSXMLParserDocumentStartError = 3,
    NSXMLParserEmptyDocumentError = 4,
    NSXMLParserPrematureDocumentEndError = 5,
    NSXMLParserInvalidHexCharacterRefError = 6,
    NSXMLParserInvalidDecimalCharacterRefError = 7,
    NSXMLParserInvalidCharacterRefError = 8,
    NSXMLParserInvalidCharacterError = 9,
    NSXMLParserCharacterRefAtEOFError = 10,
    NSXMLParserCharacterRefInPrologError = 11,
    NSXMLParserCharacterRefInEpilogError = 12,
    NSXMLParserCharacterRefInDTDError = 13,
    NSXMLParserEntityRefAtEOFError = 14,
    NSXMLParserEntityRefInPrologError = 15,
    NSXMLParserEntityRefInEpilogError = 16,
    NSXMLParserEntityRefInDTDError = 17,
    NSXMLParserParsedEntityRefAtEOFError = 18,
    NSXMLParserParsedEntityRefInPrologError = 19,
    NSXMLParserParsedEntityRefInEpilogError = 20,
    NSXMLParserParsedEntityRefInInternalSubsetError = 21,
    NSXMLParserEntityReferenceWithoutNameError = 22,
    NSXMLParserEntityReferenceMissingSemiError = 23,
    NSXMLParserParsedEntityRefNoNameError = 24,
    NSXMLParserParsedEntityRefMissingSemiError = 25,
    NSXMLParserUndeclaredEntityError = 26,
    NSXMLParserUnparsedEntityError = 28,
    NSXMLParserEntityIsExternalError = 29,
    NSXMLParserEntityIsParameterError = 30,
    NSXMLParserUnknownEncodingError = 31,
    NSXMLParserEncodingNotSupportedError = 32,
    NSXMLParserStringNotStartedError = 33,
    NSXMLParserStringNotClosedError = 34,
    NSXMLParserNamespaceDeclarationError = 35,
    NSXMLParserEntityNotStartedError = 36,
    NSXMLParserEntityNotFinishedError = 37,
    NSXMLParserLessThanSymbolInAttributeError = 38,
    NSXMLParserAttributeNotStartedError = 39,
    NSXMLParserAttributeNotFinishedError = 40,
    NSXMLParserAttributeHasNoValueError = 41,
    NSXMLParserAttributeRedefinedError = 42,
    NSXMLParserLiteralNotStartedError = 43,
    NSXMLParserLiteralNotFinishedError = 44,
    NSXMLParserCommentNotFinishedError = 45,
    NSXMLParserProcessingInstructionNotStartedError = 46,
    NSXMLParserProcessingInstructionNotFinishedError = 47,
    NSXMLParserNotationNotStartedError = 48,
    NSXMLParserNotationNotFinishedError = 49,
    NSXMLParserAttributeListNotStartedError = 50,
    NSXMLParserAttributeListNotFinishedError = 51,
    NSXMLParserMixedContentDeclNotStartedError = 52,
    NSXMLParserMixedContentDeclNotFinishedError = 53,
    NSXMLParserElementContentDeclNotStartedError = 54,
    NSXMLParserElementContentDeclNotFinishedError = 55,
    NSXMLParserXMLDeclNotStartedError = 56,
    NSXMLParserXMLDeclNotFinishedError = 57,
    NSXMLParserConditionalSectionNotStartedError = 58,

```

```

NSXMLParserConditionalSectionNotFinishedError = 59,
NSXMLParserExternalSubsetNotFinishedError = 60,
NSXMLParserDOCTYPEDeclNotFinishedError = 61,
NSXMLParserMisplacedCDATAEndStringError = 62,
NSXMLParserCDATANotFinishedError = 63,
NSXMLParserMisplacedXMLDeclarationError = 64,
NSXMLParserSpaceRequiredError = 65,
NSXMLParserSeparatorRequiredError = 66,
NSXMLParserNMTOKENRequiredError = 67,
NSXMLParserNAMERequiredError = 68,
NSXMLParserPCDATARequiredError = 69,
NSXMLParserURIRequiredError = 70,
NSXMLParserPublicIdentifierRequiredError = 71,
NSXMLParserLTRRequiredError = 72,
NSXMLParserGTRRequiredError = 73,
NSXMLParserLTSlashRequiredError = 74,
NSXMLParserEqualExpectedError = 75,
NSXMLParserTagNameMismatchError = 76,
NSXMLParserUnfinishedTagError = 77,
NSXMLParserStandaloneValueError = 78,
NSXMLParserInvalidEncodingNameError = 79,
NSXMLParserCommentContainsDoubleHyphenError = 80,
NSXMLParserInvalidEncodingError = 81,
NSXMLParserExternalStandaloneEntityError = 82,
NSXMLParserInvalidConditionalSectionError = 83,
NSXMLParserEntityValueRequiredError = 84,
NSXMLParserNotWellBalancedError = 85,
NSXMLParserExtraContentError = 86,
NSXMLParserInvalidCharacterInEntityError = 87,
NSXMLParserParsedEntityRefInInternalError = 88,
NSXMLParserEntityRefLoopError = 89,
NSXMLParserEntityBoundaryError = 90,
NSXMLParserInvalidURIError = 91,
NSXMLParserURIFragmentError = 92,
NSXMLParserNoDTDError = 94,
NSXMLParserDelegateAbortedParseError = 512
} NSXMLParserError;

```

### Constants

`NSXMLParserInternalError`

The parser object encountered an internal error.

Available in Mac OS X v10.3 and later.

Declared in `NSXMLParser.h`.

`NSXMLParserOutOfMemoryError`

The parser object ran out of memory.

Available in Mac OS X v10.3 and later.

Declared in `NSXMLParser.h`.

`NSXMLParserDocumentStartError`

The parser object is unable to start parsing.

Available in Mac OS X v10.3 and later.

Declared in `NSXMLParser.h`.

NSXMLParserEmptyDocumentError

**The document is empty.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserPrematureDocumentEndError

**The document ended unexpectedly.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserInvalidHexCharacterRefError

**Invalid hexadecimal character reference encountered.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserInvalidDecimalCharacterRefError

**Invalid decimal character reference encountered.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserInvalidCharacterRefError

**Invalid character reference encountered.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserInvalidCharacterError

**Invalid character encountered.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserCharacterRefAtEOFError

**Target of character reference cannot be found.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserCharacterRefInPrologError

**Invalid character found in the prolog.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserCharacterRefInEpilogError

**Invalid character found in the epilog.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

NSXMLParserCharacterRefInDTDError

**Invalid character encountered in the DTD.**

**Available in Mac OS X v10.3 and later.**

**Declared in NSXMLParser.h.**

- `NSXMLParserEntityRefAtEOFError`  
Target of entity reference is not found.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityRefInPrologError`  
Invalid entity reference found in the prolog.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityRefInEpilogError`  
Invalid entity reference found in the epilog.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityRefInDTDError`  
Invalid entity reference found in the DTD.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserParsedEntityRefAtEOFError`  
Target of parsed entity reference is not found.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserParsedEntityRefInPrologError`  
Target of parsed entity reference is not found in prolog.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserParsedEntityRefInEpilogError`  
Target of parsed entity reference is not found in epilog.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserParsedEntityRefInInternalSubsetError`  
Target of parsed entity reference is not found in internal subset.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityReferenceWithoutNameError`  
Entity reference is without name.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityReferenceMissingSemiError`  
Entity reference is missing semicolon.  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.

- `NSXMLParserParsedEntityRefNoNameError`  
**Parsed entity reference is without an entity name.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserParsedEntityRefMissingSemiError`  
**Parsed entity reference is missing semicolon.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserUndeclaredEntityError`  
**Entity is not declared.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserUnparsedEntityError`  
**Cannot parse entity.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityIsExternalError`  
**Cannot parse external entity.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityIsParameterError`  
**Entity is a parameter.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserUnknownEncodingError`  
**Document encoding is unknown.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEncodingNotSupportedError`  
**Document encoding is not supported.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserStringNotStartedError`  
**String is not started.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserStringNotClosedError`  
**String is not closed.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.

NSXMLParserNamespaceDeclarationError  
**Invalid namespace declaration encountered.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserEntityNotStartedError  
**Entity is not started.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserEntityNotFinishedError  
**Entity is not finished.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserLessThanSymbolInAttributeError  
**Angle bracket is used in attribute.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserAttributeNotStartedError  
**Attribute is not started.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserAttributeNotFinishedError  
**Attribute is not finished.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserAttributeHasNoValueError  
**Attribute doesn't contain a value.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserAttributeRedefinedError  
**Attribute is redefined.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserLiteralNotStartedError  
**Literal is not started.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.

NSXMLParserLiteralNotFinishedError  
**Literal is not finished.**  
Available in Mac OS X v10.3 and later.  
Declared in NSXMLParser.h.



NSXMLParserCommentNotFinishedError

**Comment is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserProcessingInstructionNotStartedError

**Processing instruction is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserProcessingInstructionNotFinishedError

**Processing instruction is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserNotationNotStartedError

**Notation is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserNotationNotFinishedError

**Notation is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserAttributeListNotStartedError

**Attribute list is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserAttributeListNotFinishedError

**Attribute list is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserMixedContentDeclNotStartedError

**Mixed content declaration is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserMixedContentDeclNotFinishedError

**Mixed content declaration is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserElementContentDeclNotStartedError

**Element content declaration is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserElementContentDeclNotFinishedError  
**Element content declaration is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserXMLDeclNotStartedError  
**XML declaration is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserXMLDeclNotFinishedError  
**XML declaration is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserConditionalSectionNotStartedError  
**Conditional section is not started.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserConditionalSectionNotFinishedError  
**Conditional section is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserExternalSubsetNotFinishedError  
**External subset is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserDOCTYPEDeclNotFinishedError  
**Document type declaration is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserMisplacedCDATAEndStringError  
**Misplaced CDATA end string.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserCDATANotFinishedError  
**CDATA block is not finished.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserMisplacedXMLDeclarationError  
**Misplaced XML declaration.**

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

- `NSXMLParserSpaceRequiredError`  
**Space is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserSeparatorRequiredError`  
**Separator is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserNMTOKENRequiredError`  
**Name token is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserNAMERequiredError`  
**Name is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserPCDATARequiredError`  
**CDATA is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserURIRequiredError`  
**URI is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserPublicIdentifierRequiredError`  
**Public identifier is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserLTRequiredError`  
**Left angle bracket is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserGTRequiredError`  
**Right angle bracket is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserLTSlashRequiredError`  
**Left angle bracket slash is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.

- `NSXMLParserEqualExpectedError`  
**Equal sign expected.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserTagNameMismatchError`  
**Tag name mismatch.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserUnfinishedTagError`  
**Unfinished tag found.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserStandaloneValueError`  
**Standalone value found.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserInvalidEncodingNameError`  
**Invalid encoding name found.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserCommentContainsDoubleHyphenError`  
**Comment contains double hyphen.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserInvalidEncodingError`  
**Invalid encoding.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserExternalStandaloneEntityError`  
**External standalone entity.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserInvalidConditionalSectionError`  
**Invalid conditional section.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.
- `NSXMLParserEntityValueRequiredError`  
**Entity value is required.**  
Available in Mac OS X v10.3 and later.  
Declared in `NSXMLParser.h`.

NSXMLParserNotWellBalancedError

Document is not well balanced.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserExtraContentError

Error in content found.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserInvalidCharacterInEntityError

Invalid character in entity found.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserParsedEntityRefInInternalError

Internal error in parsed entity reference found.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserEntityRefLoopError

Entity reference loop encountered.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserEntityBoundaryError

Entity boundary error.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserInvalidURIError

Invalid URI specified.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserURIFragmentError

URI fragment.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserNoDTDError

Missing DTD.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

NSXMLParserDelegateAbortedParseError

Delegate aborted parse.

Available in Mac OS X v10.3 and later.

Declared in NSXMLParser.h.

**Declared In**

NSXMLParser.h



# Document Revision History

---

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

Date	Notes
2007-02-02	Merged delegate method groups with regular method groups.
2006-11-07	Added version information for namespace support.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History



# Index

---

## A

---

`abortParsing` **instance method** [8](#)

## C

---

`columnNumber` **instance method** [8](#)

## D

---

`delegate` **instance method** [9](#)

## I

---

`initWithContentsOfURL:` **instance method** [9](#)

`initWithData:` **instance method** [10](#)

## L

---

`lineNumber` **instance method** [10](#)

## N

---

`NSXMLParserAttributeHasNoValueError` **constant** [32](#)

`NSXMLParserAttributeListNotFinishedError` **constant** [33](#)

`NSXMLParserAttributeListNotStartedError` **constant** [33](#)

`NSXMLParserAttributeNotFinishedError` **constant** [32](#)

`NSXMLParserAttributeNotStartedError` **constant** [32](#)

`NSXMLParserAttributeRedefinedError` **constant** [32](#)

`NSXMLParserCDATANotFinishedError` **constant** [34](#)

`NSXMLParserCharacterRefAtEOFError` **constant** [29](#)

`NSXMLParserCharacterRefInDTDError` **constant** [29](#)

`NSXMLParserCharacterRefInEpilogError` **constant** [29](#)

`NSXMLParserCharacterRefInPrologError` **constant** [29](#)

`NSXMLParserCommentContainsDoubleHyphenError` **constant** [36](#)

`NSXMLParserCommentNotFinishedError` **constant** [33](#)

`NSXMLParserConditionalSectionNotFinishedError` **constant** [34](#)

`NSXMLParserConditionalSectionNotStartedError` **constant** [34](#)

`NSXMLParserDelegateAbortedParseError` **constant** [37](#)

`NSXMLParserDOCTYPEDeclNotFinishedError` **constant** [34](#)

`NSXMLParserDocumentStartError` **constant** [28](#)

`NSXMLParserElementContentDeclNotFinishedError` **constant** [34](#)

`NSXMLParserElementContentDeclNotStartedError` **constant** [33](#)

`NSXMLParserEmptyDocumentError` **constant** [29](#)

`NSXMLParserEncodingNotSupportedError` **constant** [31](#)

`NSXMLParserEntityBoundaryError` **constant** [37](#)

`NSXMLParserEntityIsExternalError` **constant** [31](#)

`NSXMLParserEntityIsParameterError` **constant** [31](#)

`NSXMLParserEntityNotFinishedError` **constant** [32](#)

`NSXMLParserEntityNotStartedError` **constant** [32](#)

`NSXMLParserEntityRefAtEOFError` **constant** [30](#)

`NSXMLParserEntityReferenceMissingSemiError` **constant** [30](#)

`NSXMLParserEntityReferenceWithoutNameError` **constant** [30](#)

`NSXMLParserEntityRefInDTDError` **constant** [30](#)

`NSXMLParserEntityRefInEpilogError` **constant** [30](#)

`NSXMLParserEntityRefInPrologError` **constant** [30](#)

`NSXMLParserEntityRefLoopError` **constant** [37](#)

- NSXMLParserEntityValueRequiredError **constant** [36](#)
- NSXMLParserEqualExpectedError **constant** [36](#)
- NSXMLParserError **data type** [25](#)
- NSXMLParserErrorDomain** [25](#)
- NSXMLParserErrorDomain **constant** [25](#)
- NSXMLParserExternalStandaloneEntityError **constant** [36](#)
- NSXMLParserExternalSubsetNotFinishedError **constant** [34](#)
- NSXMLParserExtraContentError **constant** [37](#)
- NSXMLParserGTRequiredError **constant** [35](#)
- NSXMLParserInternalError **constant** [28](#)
- NSXMLParserInvalidCharacterError **constant** [29](#)
- NSXMLParserInvalidCharacterInEntityError **constant** [37](#)
- NSXMLParserInvalidCharacterRefError **constant** [29](#)
- NSXMLParserInvalidConditionalSectionError **constant** [36](#)
- NSXMLParserInvalidDecimalCharacterRefError **constant** [29](#)
- NSXMLParserInvalidEncodingError **constant** [36](#)
- NSXMLParserInvalidEncodingNameError **constant** [36](#)
- NSXMLParserInvalidHexCharacterRefError **constant** [29](#)
- NSXMLParserInvalidURIError **constant** [37](#)
- NSXMLParserLessThanSymbolInAttributeError **constant** [32](#)
- NSXMLParserLiteralNotFinishedError **constant** [32](#)
- NSXMLParserLiteralNotStartedError **constant** [32](#)
- NSXMLParserLTRequiredError **constant** [35](#)
- NSXMLParserLTSlashRequiredError **constant** [35](#)
- NSXMLParserMisplacedCDATAEndStringError **constant** [34](#)
- NSXMLParserMisplacedXMLDeclarationError **constant** [34](#)
- NSXMLParserMixedContentDeclNotFinishedError **constant** [33](#)
- NSXMLParserMixedContentDeclNotStartedError **constant** [33](#)
- NSXMLParserNAMERequiredError **constant** [35](#)
- NSXMLParserNamespaceDeclarationError **constant** [32](#)
- NSXMLParserNMTOKENRequiredError **constant** [35](#)
- NSXMLParserNoDTDError **constant** [37](#)
- NSXMLParserNotationNotFinishedError **constant** [33](#)
- NSXMLParserNotationNotStartedError **constant** [33](#)
- NSXMLParserNotWellBalancedError **constant** [37](#)
- NSXMLParserOutOfMemoryError **constant** [28](#)
- NSXMLParserParsedEntityRefAtEOFError **constant** [30](#)
- NSXMLParserParsedEntityRefInEpilogError **constant** [30](#)
- NSXMLParserParsedEntityRefInInternalError **constant** [37](#)
- NSXMLParserParsedEntityRefInInternalSubsetError **constant** [30](#)
- NSXMLParserParsedEntityRefInPrologError **constant** [30](#)
- NSXMLParserParsedEntityRefMissingSemiError **constant** [31](#)
- NSXMLParserParsedEntityRefNoNameError **constant** [31](#)
- NSXMLParserPCDATARequiredError **constant** [35](#)
- NSXMLParserPrematureDocumentEndError **constant** [29](#)
- NSXMLParserProcessingInstructionNotFinishedError **constant** [33](#)
- NSXMLParserProcessingInstructionNotStartedError **constant** [33](#)
- NSXMLParserPublicIdentifierRequiredError **constant** [35](#)
- NSXMLParserSeparatorRequiredError **constant** [35](#)
- NSXMLParserSpaceRequiredError **constant** [35](#)
- NSXMLParserStandaloneValueError **constant** [36](#)
- NSXMLParserStringNotClosedError **constant** [31](#)
- NSXMLParserStringNotStartedError **constant** [31](#)
- NSXMLParserTagNameMismatchError **constant** [36](#)
- NSXMLParserUndeclaredEntityError **constant** [31](#)
- NSXMLParserUnfinishedTagError **constant** [36](#)
- NSXMLParserUnknownEncodingError **constant** [31](#)
- NSXMLParserUnparsedEntityError **constant** [31](#)
- NSXMLParserURIFragmentError **constant** [37](#)
- NSXMLParserURIRequiredError **constant** [35](#)
- NSXMLParserXMLDeclNotFinishedError **constant** [34](#)
- NSXMLParserXMLDeclNotStartedError **constant** [34](#)

## P

---

parse **instance method** [10](#)

**Parser Error Constants** [26](#)

parser:didEndElement:namespaceURI:qualifiedName:  
<NSObject> **delegate method** [15](#)

parser:didEndMappingPrefix: <NSObject> **delegate  
method** [16](#)

parser:didStartElement:namespaceURI:qualifiedName:  
attributes: <NSObject> **delegate method** [16](#)

parser:didStartMappingPrefix:toURI: <NSObject>  
**delegate method** [17](#)

parser:foundAttributeDeclarationWithName:  
   forElement:type:defaultValue: <NSObject>  
   **delegate method 17**  
 parser:foundCDATA: <NSObject> **delegate method 18**  
 parser:foundCharacters: <NSObject> **delegate method 18**  
 parser:foundComment: <NSObject> **delegate method 19**  
 parser:foundElementDeclarationWithName:model:  
   <NSObject> **delegate method 19**  
 parser:foundExternalEntityDeclarationWithName:  
   publicID:systemID: <NSObject> **delegate method 19**  
 parser:foundIgnorableWhitespace: <NSObject>  
   **delegate method 20**  
 parser:foundInternalEntityDeclarationWithName:  
   value: <NSObject> **delegate method 21**  
 parser:foundNotationDeclarationWithName:publicID:  
   systemID: <NSObject> **delegate method 21**  
 parser:foundProcessingInstructionWithTarget:data:  
   <NSObject> **delegate method 22**  
 parser:foundUnparsedEntityDeclarationWithName:  
   publicID:systemID:notationName: <NSObject>  
   **delegate method 22**  
 parser:parseErrorOccurred: <NSObject> **delegate method 23**  
 parser:resolveExternalEntityName:systemID:  
   <NSObject> **delegate method 23**  
 parser:validationErrorOccurred: <NSObject>  
   **delegate method 24**  
 parserDidEndDocument: <NSObject> **delegate method 24**  
 parserDidStartDocument: <NSObject> **delegate method 25**  
 parserError **instance method 11**  
 publicID **instance method 11**

## S

---

setDelegate: **instance method 11**  
 setShouldProcessNamespaces: **instance method 12**  
 setShouldReportNamespacePrefixes: **instance method 12**  
 setShouldResolveExternalEntities: **instance method 13**  
 shouldProcessNamespaces **instance method 13**  
 shouldReportNamespacePrefixes **instance method 14**  
 shouldResolveExternalEntities **instance method 14**  
 systemID **instance method 15**