# NSMethodSignature Class Reference

**Cocoa > Objective-C Language**

**2008-10-15**

Apple Inc.
© 2008 Apple Inc.
All rights reserved.

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

The Apple logo is a trademark of Apple Inc.

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

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

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

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

Apple, the Apple logo, Cocoa, Mac, Mac OS, and Objective-C 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


## NSMethodSignature Class Reference 5

Overview 5
Tasks 6
- Creating a Method Signature Object 6
- Getting Information on Argument Types 6
- Getting Information on Return Types 6
- Determining Synchronous Status 6

Class Methods 6
- signatureWithObjCTypes: 6

Instance Methods 7
- frameLength 7
- getArgumentTypeAtIndex: 7
- isOneway 8
- methodReturnLength 8
- methodReturnType 9
- numberOfArguments 9

## Document Revision History 11

## Index 13




2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

# NSMethodSignature 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** | NSMethodSignature.h |
| **Companion guides** | Distributed Objects Programming Topics<br>The Objective-C 2.0 Programming Language |

## Overview

An `NSMethodSignature` object records type information for the arguments and return value of a method. It is used to forward messages that the receiving object does not respond to—most notably in the case of distributed objects. You typically create an `NSMethodSignature` object using `NSObject`'s `methodSignatureForSelector:` instance method (on Mac OS X v10.5 and later you can also use `signatureWithObjCTypes:` (page 6)). It is then used to create an `NSInvocation` object, which is passed as the argument to a `forwardInvocation:` message to send the invocation on to whatever other object can handle the message. In the default case, `NSObject` invokes `doesNotRecognizeSelector:`, which raises an exception. For distributed objects, the `NSInvocation` object is encoded using the information in the `NSMethodSignature` object and sent to the real object represented by the receiver of the message.

An `NSMethodSignature` object presents its argument types by index with the `getArgumentTypeAtIndex:` (page 7) method. The hidden arguments for every method, *`self`* and *`_cmd`*, are at indices `0` and `1`, respectively. The arguments normally specified in a message invocation follow these. In addition to the argument types, an `NSMethodSignature` object offers the total number of arguments with `numberOfArguments` (page 9), the total stack frame length occupied by all arguments with `frameLength` (page 7) (this varies with hardware architecture), and the length and type of the return value with `methodReturnLength` (page 8) and `methodReturnType` (page 9). Finally, applications using distributed objects can determine if the method is asynchronous with the `isOneway` (page 8) method.

For more information about the nature of a method, including the hidden arguments, see "How Messaging Works" in *The Objective-C 2.0 Programming Language*.


2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

# Tasks

## Creating a Method Signature Object

`+ signatureWithObjCTypes:` (page 6)
Returns an `NSMethodSignature` object for the given Objective C method type string.

## Getting Information on Argument Types

`- getArgumentTypeAtIndex:` (page 7)
Returns the type encoding for the argument at a given index.

`- numberOfArguments` (page 9)
Returns the number of arguments recorded in the receiver.

`- frameLength` (page 7)
Returns the number of bytes that the arguments, taken together, occupy on the stack.

## Getting Information on Return Types

`- methodReturnType` (page 9)
Returns a C string encoding the return type of the method in Objective-C type encoding.

`- methodReturnLength` (page 8)
Returns the number of bytes required for the return value.

## Determining Synchronous Status

`- isOneway` (page 8)
Returns a Boolean value that indicates whether the receiver is asynchronous when invoked through distributed objects.

# Class Methods

## signatureWithObjCTypes:

Returns an `NSMethodSignature` object for the given Objective C method type string.

```
+ (NSMethodSignature *)signatureWithObjCTypes:(const char *)types
```

**Parameters**

*types*
An array of characters containing the type encodings for the method arguments.

Indices begin with `0`. The hidden arguments *self* (of type `id`) and *_cmd* (of type SEL) are at indices `0` and `1`; method-specific arguments begin at index `2`.


2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

**Return Value**
An `NSMethodSignature` object for the given Objective C method type string in *types*.

**Discussion**

**Special Considerations**

This method, available since Mac OS X v10.0, is exposed in Mac OS X v10.5. Only type encoding strings of the style of the runtime that the application is running against are supported. In exposing this method there is no commitment to binary compatibily supporting any "old-style" type encoding strings after such changes occur.

It is your responsibility to pass in type strings which are either from the current runtime data or match the style of type string in use by the runtime that the application is running on.

**Availability**
Available in Mac OS X v10.5 and later.

**Declared In**
`NSMethodSignature.h`

# Instance Methods

## frameLength

Returns the number of bytes that the arguments, taken together, occupy on the stack.

```
- (NSUInteger)frameLength
```

**Return Value**
The number of bytes that the arguments, taken together, occupy on the stack.

**Discussion**
This number varies with the hardware architecture the application runs on.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`NSMethodSignature.h`

## getArgumentTypeAtIndex:

Returns the type encoding for the argument at a given index.

```
- (const char *)getArgumentTypeAtIndex:(NSUInteger)index
```

**Parameters**

*index*

The index of the argument to get.

**Return Value**
The type encoding for the argument at *index*.


2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

**Discussion**
Indices begin with `0`. The hidden arguments *`self`* (of type `id`) and *`_cmd`* (of type SEL) are at indices `0` and `1`; method-specific arguments begin at index `2`. Raises `NSInvalidArgumentException` if *`index`* is too large for the actual number of arguments.

Argument types are given as C strings with Objective-C type encoding. This encoding is implementation-specific, so applications should use it with caution.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`NSMethodSignature.h`

## isOneway

Returns a Boolean value that indicates whether the receiver is asynchronous when invoked through distributed objects.

```
- (BOOL)isOneway
```

**Return Value**
`YES` if the receiver is asynchronous when invoked through distributed objects, otherwise `NO`.

**Discussion**
If the method is `oneway`, the sender of the remote message doesn't block awaiting a reply.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`NSMethodSignature.h`

## methodReturnLength

Returns the number of bytes required for the return value.

```
- (NSUInteger)methodReturnLength
```

**Return Value**
The number of bytes required for the return value.

**Availability**
Available in Mac OS X v10.0 and later.

**See Also**
`- methodReturnType` (page 9)

**Declared In**
`NSMethodSignature.h`


2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

## methodReturnType

Returns a C string encoding the return type of the method in Objective-C type encoding.

```
- (const char *)methodReturnType
```

**Return Value**
A C string encoding the return type of the method in Objective-C type encoding.

**Discussion**
This encoding is implementation-specific, so applications should use it with caution.

**Availability**
Available in Mac OS X v10.0 and later.

**See Also**
- methodReturnLength (page 8)

**Declared In**
NSMethodSignature.h

## numberOfArguments

Returns the number of arguments recorded in the receiver.

```
- (NSUInteger)numberOfArguments
```

**Return Value**
The number of arguments recorded in the receiver.

**Discussion**
There are always at least 2 arguments, because an NSMethodSignature object includes the hidden arguments *self* and *_cmd*, which are the first two arguments passed to every method implementation.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
NSMethodSignature.h


2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

# Document Revision History

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

| Date | Notes |
| --- | --- |
| 2008-10-15 | Added declaration of signatureWithObjCTypes:. |
| 2008-03-11 | Added a live link to a cross-reference in the Overview section. |
| 2007-02-23 | Revised task heading. |
| 2006-05-23 | First publication of this content as a separate document. |



2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

Document Revision History



2008-10-15 | © 2008 Apple Inc. All Rights Reserved.

# Index


## F

`frameLength` instance method 7

## G

`getArgumentTypeAtIndex:` instance method 7

## I

`isOneway` instance method 8

## M

`methodReturnLength` instance method 8
`methodReturnType` instance method 9

## N

`numberOfArguments` instance method 9

## S

`signatureWithObjCTypes:` class method 6




2008-10-15 | © 2008 Apple Inc. All Rights Reserved.