

---

# NSInvocationOperation Class Reference

[Cocoa](#) > [Interapplication Communication](#)





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

---

## **NSInvocationOperation Class Reference 5**

---

Overview	5
Tasks	5
Initialization	5
Getting Attributes	5
Instance Methods	6
initWithInvocation:	6
initWithTarget:selector:object:	6
invocation	7
result	7
Constants	7
Result Exceptions	7

---

## **Document Revision History 9**

---

## **Index 11**

---



# NSInvocationOperation Class Reference

---

<b>Inherits from</b>	NSOperation : NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Threading Programming Guide
<b>Declared in</b>	NSOperation.h

## Overview

The `NSInvocationOperation` class is a concrete subclass of `NSOperation` that manages the execution of a single encapsulated task specified as an invocation. You can use this class to initiate an operation that consists of invoking a selector on a specified object. This class implements a non-concurrent operation.

For more information on concurrent versus non-concurrent operations, see *NSOperation Class Reference*.

## Tasks

### Initialization

- [initWithTarget:selector:object:](#) (page 6)  
Returns an `NSInvocationOperation` object initialized with the specified target and selector.
- [initWithInvocation:](#) (page 6)  
Returns an `NSInvocationOperation` object initialized with the specified invocation object.

### Getting Attributes

- [invocation](#) (page 7)  
Returns the receiver's invocation object.
- [result](#) (page 7)  
Returns the result of the invocation or method.

## Instance Methods

### **initWithInvocation:**

Returns an `NSInvocationOperation` object initialized with the specified invocation object.

```
- (id)initWithInvocation:(NSInvocation *)inv
```

#### **Parameters**

*inv*

The invocation object identifying the target object, selector, and parameter objects.

#### **Return Value**

An initialized `NSInvocationOperation` object or `nil` if the object could not be initialized.

#### **Discussion**

This method is the designated initializer. The receiver tells the invocation object to retain its arguments.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`NSOperation.h`

### **initWithTarget:selector:object:**

Returns an `NSInvocationOperation` object initialized with the specified target and selector.

```
- (id)initWithTarget:(id)target selector:(SEL)sel object:(id)arg
```

#### **Parameters**

*target*

The object defining the specified selector.

*sel*

The selector to invoke when running the operation. The selector may take 0 or 1 parameters. If it accepts a parameter, the type of that parameter should be `id`.

*arg*

The parameter object to pass to the selector. If the selector does not take an argument, specify `nil`.

#### **Return Value**

An initialized `NSInvocationOperation` object or `nil` if the target object does not implement the specified selector.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`NSOperation.h`

## invocation

Returns the receiver's invocation object.

- (NSInvocation \*)invocation

### Return Value

The invocation object identifying the target object, selector, and parameters to use to execute the operation's task.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [initWithTarget:selector:object:](#) (page 6)
- [initWithInvocation:](#) (page 6)

### Declared In

NSOperation.h

## result

Returns the result of the invocation or method.

- (id)result

### Return Value

The object returned by the method or an `NSValue` object containing the return value if it is not an object. If the method or invocation is not finished executing, this method returns `nil`.

### Discussion

If an exception was raised during the execution of the method or invocation, this method raises that exception again. If the operation was cancelled or the invocation or method has a `void` return type, calling this method raises an exception; see “[Result Exceptions](#)” (page 7).

### Availability

Available in Mac OS X v10.5 and later.

### Declared In

NSOperation.h

## Constants

### Result Exceptions

Names of exceptions raised by `NSInvocationOperation` if there is an error when calling the `result` (page 7) method.

```
extern NSString * const NSInvocationOperationVoidResultException;  
extern NSString * const NSInvocationOperationCancelledException;
```

### Constants

`NSInvocationOperationVoidResultException`

The name of the exception raised if the `result` method is called for an invocation method with a `void` return type.

Available in Mac OS X v10.5 and later.

Declared in `NSOperation.h`.

`NSInvocationOperationCancelledException`

The name of the exception raised if the `result` method is called after the operation was cancelled.

Available in Mac OS X v10.5 and later.

Declared in `NSOperation.h`.

### Declared In

`NSOperation.h`



# Document Revision History

---

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

Date	Notes
2007-04-12	New document describing an invocation-based implementation of the NSOperation class.

## REVISION HISTORY

### Document Revision History

# Index

---

## I

---

`initWithInvocation:` **instance method** [6](#)  
`initWithTarget:selector:object:` **instance method**  
[6](#)  
`invocation` **instance method** [7](#)

## N

---

`NSInvocationOperationCancelledException`  
**constant** [8](#)  
`NSInvocationOperationVoidResultException`  
**constant** [8](#)

## R

---

**Result Exceptions** [7](#)  
`result` **instance method** [7](#)