

---

# NSAssertionHandler Class Reference

[Cocoa](#) > [Performance](#)



2007-02-22



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

---

## **NSAssertionHandler Class Reference 5**

---

Overview 5

Tasks 5

    Handling Assertion Failures 5

Class Methods 6

    currentHandler 6

Instance Methods 6

    handleFailureInFunction:file:lineNumber:description: 6

    handleFailureInMethod:object:file:lineNumber:description: 7

---

## **Document Revision History 9**

---

## **Index 11**

---



# NSAssertionHandler 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 v10.0 and later.
<b>Companion guide</b>	Assertions and Logging
<b>Declared in</b>	NSExcption.h

## Overview

`NSAssertionHandler` objects are automatically created to handle false assertions. Assertion macros, such as `NSAssert` and `NSCAssert`, are used to evaluate a condition, and, if the condition evaluates to false, the macros pass a string to an `NSAssertionHandler` object describing the failure. Each thread has its own `NSAssertionHandler` object. When invoked, an assertion handler prints an error message that includes the method and class (or function) containing the assertion and raises an `NSInternalInconsistencyException`.

You create assertions only using the assertion macros—you rarely need to invoke `NSAssertionHandler` methods directly. The macros for use inside methods and functions send `handleFailureInMethod:object:file:lineNumber:description:` (page 7) and `handleFailureInFunction:file:lineNumber:description:` (page 6) messages respectively to the current assertion handler. The assertion handler for the current thread is obtained using the `currentHandler` (page 6) class method. If you need to customize the behavior of `NSAssertionHandler`, create a subclass, overriding the above two methods, and install your instance into the current thread's attributes dictionary with the key `NSAssertionHandler`.

## Tasks

### Handling Assertion Failures

+ `currentHandler` (page 6)

Returns the `NSAssertionHandler` object associated with the current thread.

- `handleFailureInFunction:file:lineNumber:description:` (page 6)

Logs (using `NSLog`) an error message that includes the name of the function, the name of the file, and the line number.

- [handleFailureInMethod:object:file:lineNumber:description:](#) (page 7)

Logs (using NSLog) an error message that includes the name of the method that failed, the class name of the object, the name of the source file, and the line number.

## Class Methods

### currentHandler

Returns the `NSAssertionHandler` object associated with the current thread.

```
+ (NSAssertionHandler *)currentHandler
```

#### Return Value

The `NSAssertionHandler` object associated with the current thread.

#### Discussion

If no assertion handler is associated with the current thread, this method creates one and assigns it to the thread.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSException.h`

## Instance Methods

### handleFailureInFunction:file:lineNumber:description:

Logs (using NSLog) an error message that includes the name of the function, the name of the file, and the line number.

```
- (void)handleFailureInFunction:(NSString *)functionName file:(NSString *)fileName
    lineNumber:(NSInteger)line description:(NSString *)format, ...
```

#### Parameters

*functionName*

The function that failed.

*object*

The object that failed.

*fileName*

The name of the source file.

*line*

The line in which the failure occurred.

*format, ...*

A format string followed by a comma-separated list of arguments to substitute into the format string. See Formatting String Objects for more information.

#### Discussion

Raises `NSInternalInconsistencyException`.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSException.h`

### **handleFailureInMethod:object:file:lineNumber:description:**

Logs (using `NSLog`) an error message that includes the name of the method that failed, the class name of the object, the name of the source file, and the line number.

```
- (void)handleFailureInMethod:(SEL)selector object:(id)object file:(NSString *)fileName lineNumber:(NSInteger)line description:(NSString *)format, ...
```

#### Parameters

*selector*

The selector for the method that failed

*object*

The object that failed.

*fileName*

The name of the source file.

*line*

The line in which the failure occurred.

*format, ...*

A format string followed by a comma-separated list of arguments to substitute into the format string. See Formatting String Objects for more information.

#### Discussion

Raises `NSInternalInconsistencyException`.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSException.h`





# Document Revision History

---

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

Date	Notes
2007-02-22	Revised task groups.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

currentHandler **class method** [6](#)

## H

---

handleFailureInFunction:file:lineNumber:  
description: **instance method** [6](#)

handleFailureInMethod:object:file:lineNumber:  
description: **instance method** [7](#)