

NSAssertionHandler

Inherits From: NSObject

Declared In: foundation/NSException.h

Class Description

Assertion failures offer a convenient way to check for assertions. The `NSAssertionHandler` class manages assertion failures. Each thread has its own assertion-handler object, which usually work in coordination with a group of assertion macros. These macros evaluate a condition and, if the condition evaluates to false, call the assertion handler for the current thread, passing it a format string and a variable number of arguments.

The assertion macros come in two sorts, some for Objective-C methods and others for C functions. Two methods of `NSAssertionHandler` mirror this division;

`handleFailureInMethod:object:file:lineNumber:description:` is used for assertion failures occurring within methods, while

`handleFailureInFunction:file:lineNumber:description` handles assertion failures occurring in C functions. Both methods begin by logging an error message with **`NSLog()`**; the only difference in the message is method selector and object

(**`handleFailureInMethod:...`**), and function name (**`handleFailureInFunction:...`**). Then both methods raise an exception of **`NSInternalInconsistencyException`**.

Usually for assertions, you need only use the assertion macros **`NSAssert()`**, **`NSCAssert()`**, and other related macros, which incorporate the appropriate `NSAssertionHandler` methods.

Instance Variables

None declared in this class.

Method Types

Getting the thread's handler + `currentHandler`

Handling assertion failures – `handleFailureInFunction:file:lineNumber:description:`
 – `handleFailureInMethod:object:file:`
 `lineNumber:description:`

Class Methods

currentHandler

+ (NSAssertionHandler *)**currentHandler**

Returns the assertion handler for the current thread.

Instance Methods

handleFailureInFunction:file:lineNumber:description

– (void)**handleFailureInFunction:**(NSString *)*functionName*
 file:(NSString *)*fileName*
 lineNumber:(int)*line*
 description:(NSString *)*formattedString*,
 ...

This method logs an error message that includes the function, file, and line number and then raises an exception.

handleFailureInMethod:object:file:lineNumber:description

– (void)**handleFailureInMethod:**(SEL)*methodSelector*
 object:*anObject*
 file:(NSString *)*fileName*
 lineNumber:(int)*line*
 description:(NSString *)*formattedString*,
 ...

This method logs an error message that includes the class, method, file, and line number and then raises an exception.