NSAssertionHandler

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSException.h

Class Description

NSAssertionHandler objects are automatically created to handle false assertions. Assertion macros 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 NSAssertionHandler prints an error message that includes the method and class (or function) containing the assertion, and raises an NSInternalInconsistencyException.

You use an assortment of macros to evaluate a condition—these macros serve as a front end to NSAssertionHandler. These macros fall into two categories: those you use in Objective-C methods, and those you use in C functions. For example, **NSAssert()** is for use within methods and **NSCAssert()** is for use within functions. Each macro has two arguments: the condition, an expression that evaluates to true or false, and the NSString describing the failure. Other macros are available if more arguments are needed for a **printf**()-style description. For example, **NSAssert1()** is used within methods if an additional argument is needed as in:

```
NSAssert1((0 <= component) && (component <= 255),
    @"Value %i out of range!", component);
```

For more details on these macros see **NSAssert()**.

You create assertions only using the above macros—you rarely need to invoke NSAssertionHandler methods directly. The macros for use inside methods and functions will send **handleFailureInMethod:...** and **handleFailureInFunction:...** to the current assertion handler respectively. The assertion handler for the current thread is obtained using the **currentHandler** class method.

Assertions are *not* compiled into code if the preprocessor macro NS_BLOCK_ASSERTIONS is defined.

Method Types

Getting the thread's handler	+ currentHandler
Handling assertion failures	– handleFailureInFunction:file:lineNumber:description:
	– handleFailureInMethod:object:file:lineNumber:description:

currentHandler

+ (NSAssertionHandler *)**currentHandler**

Returns an NSAssertionHandler for the current thread.

handleFailureInFunction:file:lineNumber:description:

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

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

handleFailureInMethod:object:file:lineNumber:description:

– (void)**handleFailureInMethod:**(SEL)*selector*
    object:(id)*object*
    file:(NSString *)*fileName*
    lineNumber:(int)*line*
    description:(NSString *)*format,...*

Logs an error message (using **NSLog()**) that includes the *selector*, *object*'s class name, the name of the file *fileName*, and the line number *line*. Raises NSInternalInconsistencyException.