NSThread

Inherits From:	NSObject
Conforms To:	NSLocking NSObject (NSObject)
Declared In:	foundation/NSThread.h

Class Description

An NSThread object controls a thread of execution. You use an NSThread when you want to terminate or delay a thread or when you want a new thread.

A *thread* is an executable unit. A *task* is made up of one or more threads. Each thread has its own execution stack and is capable of independent I/O. All threads share the virtual memory address space and communication rights of their task. When a thread is started, it is *detached* from its initiating thread. The new thread runs independently. That is, the initiating thread does not know the new thread's state.

To obtain an NSThread object that represents your current thread of execution, use the **currentThread** method. To obtain an NSThread object that will create a new thread of execution, use **detachNewThreadSelector:toTarget:withObject:**. This method sends the specified Objective C message to the specified object in its own thread of execution. You use the NSThread object returned by these methods if you ever need to delay or terminate that thread of execution.

When you use detachNewThreadSelector:toTarget:withObject:, your application becomes multithreaded. At any time, you can send isMultiThreaded to find out if the application is multithreaded, that is, if a thread was ever detached from the current thread. isMultiThreaded returns YES even if the detached thread has completed execution.

Method Types

Creating an NSThread	+ currentThread + detachNewThreadSelector:toTarget:withObject:
Querying a Thread	+ isMultiThreaded - threadDictionary

Terminating a Thread - exit

Class Methods

currentThread

+ (NSThread *)**currentThread**

Returns an object representing the current thread of execution.

detachNewThreadSelector:toTarget:withObject:

+ (void)**detachNewThreadSelector:**(SEL)*aSelector* **toTarget:**(id)*aTarget* **withObject:**(id)*anArgument*

Creates and starts a new NSThread for the message **[*aTarget aSelector*:*anArgument*]**. The method *aSelector* may take only one argument and may not have a return value. If this is the first thread detached from the current thread, this method posts the notification NSBecomingMultiThreaded will the nil object to the default notification center.

isMultiThreaded

+ (BOOL)**isMultiThreaded**

Returns YES if a thread was ever detached (regardless of if the detached thread is still running).

Instance Methods

exit

– (void)**exit**

Terminates the thread represented by the calling object. Before exiting that thread, this method posts the NSThreadExiting notification with the thread being exited to the default notification center.

threadDictionary

– (NSMutableDictionary *)**threadDictionary**

Returns the NSThread's dictionary, allowing you to add data specific to the receiving NSThread. This essentially allows user-defined NSThread variables.