NSLock Class Reference

Cocoa > Process Management

2008-02-08

Contents

NSLock Class Reference

Inherits from	NSObject
Conforms to	NSLocking NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Threading Programming Guide
Declared in	NSLock.h
Related sample code	Aperture Image Resizer ExtractMovieAudioToAIFF QTExtractAndConvertToAIFF QTQuartzPlayer SimpleThreads

Overview

An `NSLock` object is used to coordinate the operation of multiple threads of execution within the same application. An `NSLock` object can be used to mediate access to an application's global data or to protect a critical section of code, allowing it to run atomically.

Warning: The `NSLock` class uses POSIX threads to implement its locking behavior. When sending an unlock message to an `NSLock` object, you must be sure that message is sent from the same thread that sent the initial lock message. Unlocking a lock from a different thread can result in undefined behavior.

You should not use this class to implement a recursive lock. Calling the `lock` method twice on the same thread will lock up your thread permanently. Use the `NSRecursiveLock` class to implement recursive locks instead.

Unlocking a lock that is not locked is considered a programmer error and should be fixed in your code. The `NSLock` class reports such errors by printing an error message to the console when they occur.

Adopted Protocols

NSLocking
- `lock`
- `unlock`

Tasks

Acquiring a Lock

`– lockBeforeDate:` (page 6)
: Attempts to acquire a lock before a given time and returns a Boolean value indicating whether the attempt was successful.

`– tryLock` (page 7)
: Attempts to acquire a lock and immediately returns a Boolean value that indicates whether the attempt was successful.

Naming the Lock

`– setName:` (page 7)
: Assigns a name to the receiver.

`– name` (page 7)
: Returns the name associated with the receiver.

Instance Methods

lockBeforeDate:

Attempts to acquire a lock before a given time and returns a Boolean value indicating whether the attempt was successful.

```
- (BOOL)lockBeforeDate:(NSDate *)limit
```

Parameters

`limit`
: The time limit for attempting to acquire a lock.

Return Value

`YES` if the lock is acquired before *`limit`*, otherwise `NO`.

Discussion

The thread is blocked until the receiver acquires the lock or *`limit`* is reached.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSLock.h`

name

Returns the name associated with the receiver.

```
- (NSString *)name
```

Return Value
The name of the receiver.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setName:` (page 7)

Declared In
`NSLock.h`

setName:

Assigns a name to the receiver.

```
- (void)setName:(NSString *)newName
```

Parameters

`newName`
The new name for the receiver. This method makes a copy of the specified string.

Discussion
You can use a name string to identify a lock within your code. Cocoa also uses this name as part of any error descriptions involving the receiver.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- name` (page 7)

Declared In
`NSLock.h`

tryLock

Attempts to acquire a lock and immediately returns a Boolean value that indicates whether the attempt was successful.

```
- (BOOL)tryLock
```

Return Value
`YES` if the lock was acquired, otherwise `NO`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSLock.h`

Document Revision History

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

Date	Notes
2008-02-08	Added a warning describing what happens when you unlock a lock that is not currently locked.
2007-05-04	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index