
NSConditionLock Class Reference

[Cocoa](#) > [Process Management](#)





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, the Apple logo, Cocoa, Mac, and Mac OS 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

NSConditionLock Class Reference 5

| | |
|--|----|
| Overview | 5 |
| Adopted Protocols | 5 |
| Tasks | 5 |
| Initializing an NSConditionLock Object | 5 |
| Returning the Condition | 6 |
| Acquiring and Releasing a Lock | 6 |
| Accessor Methods | 6 |
| Instance Methods | 6 |
| condition | 6 |
| initWithCondition: | 7 |
| lockBeforeDate: | 7 |
| lockWhenCondition: | 7 |
| lockWhenCondition:beforeDate: | 8 |
| name | 8 |
| setName: | 9 |
| tryLock | 9 |
| tryLockWhenCondition: | 10 |
| unlockWithCondition: | 10 |

Document Revision History 11

Index 13

NSConditionLock 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 | QTAudioExtractionPanel Vertex Optimization |

Overview

The `NSConditionLock` class defines objects whose locks can be associated with specific, user-defined conditions. Using an `NSConditionLock` object, you can ensure that a thread can acquire a lock only if a certain condition is met. Once it has acquired the lock and executed the critical section of code, the thread can relinquish the lock and set the associated condition to something new. The conditions themselves are arbitrary: you define them as needed for your application.

Adopted Protocols

NSLocking
lock
unlock

Tasks

Initializing an NSConditionLock Object

- [initWithCondition:](#) (page 7)
Initializes a newly allocated `NSConditionLock` object and sets its condition.

Returning the Condition

- `condition` (page 6)
Returns the condition associated with the receiver.

Acquiring and Releasing a Lock

- `lockBeforeDate:` (page 7)
Attempts to acquire a lock before a specified moment in time.
- `lockWhenCondition:` (page 7)
Attempts to acquire a lock.
- `lockWhenCondition:beforeDate:` (page 8)
Attempts to acquire a lock before a specified moment in time.
- `tryLock` (page 9)
Attempts to acquire a lock without regard to the receiver's condition.
- `tryLockWhenCondition:` (page 10)
Attempts to acquire a lock if the receiver's condition is equal to the specified condition.
- `unlockWithCondition:` (page 10)
Relinquishes the lock and sets the receiver's condition.

Accessor Methods

- `setName:` (page 9)
Assigns a name to the receiver.
- `name` (page 8)
Returns the name associated with the receiver.

Instance Methods

condition

Returns the condition associated with the receiver.

- `(NSInteger)condition`

Return Value

The condition associated with the receiver. If no condition has been set, returns 0.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSLock.h

initWithCondition:

Initializes a newly allocated `NSConditionLock` object and sets its condition.

```
- (id)initWithCondition:(NSInteger)condition
```

Parameters

condition

The user-defined condition for the lock. The value of *condition* is user-defined; see the class description for more information.

Return Value

An initialized condition lock object; may be different than the original receiver.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTAudioExtractionPanel

Vertex Optimization

Declared In

NSLock.h

lockBeforeDate:

Attempts to acquire a lock before a specified moment in time.

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

Parameters

limit

The date by which the lock must be acquired or the attempt will time out.

Return Value

YES if the lock is acquired within the time limit, NO otherwise.

Discussion

The condition associated with the receiver isn't taken into account in this operation. This method blocks the thread's execution until the receiver acquires the lock or *limit* is reached.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lockWhenCondition:beforeDate:](#) (page 8)

Declared In

NSLock.h

lockWhenCondition:

Attempts to acquire a lock.

- (void)lockWhenCondition:(NSInteger)*condition*

Parameters

condition

The condition to match on.

Discussion

The receiver's condition must be equal to *condition* before the locking operation will succeed. This method blocks the thread's execution until the lock can be acquired.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lockWhenCondition:beforeDate:](#) (page 8)
- [unlockWithCondition:](#) (page 10)

Declared In

NSLock.h

lockWhenCondition:beforeDate:

Attempts to acquire a lock before a specified moment in time.

- (BOOL)lockWhenCondition:(NSInteger)*condition* beforeDate:(NSDate *)*limit*

Parameters

condition

The condition to match on.

limit

The date by which the lock must be acquired or the attempt will time out.

Return Value

YES if the lock is acquired within the time limit, NO otherwise.

Discussion

The receiver's condition must be equal to *condition* before the locking operation will succeed. This method blocks the thread's execution until the lock can be acquired or *limit* is reached.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lockBeforeDate:](#) (page 7)
- [lockWhenCondition:](#) (page 7)

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 9)

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 condition 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 8)

Declared In

NSLock.h

tryLock

Attempts to acquire a lock without regard to the receiver's condition.

- (BOOL)tryLock

Return Value

YES if the lock could be acquired, NO otherwise.

Discussion

This method returns immediately.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [tryLockWhenCondition:](#) (page 10)

Declared In

NSLock.h

tryLockWhenCondition:

Attempts to acquire a lock if the receiver's condition is equal to the specified condition.

- (BOOL)tryLockWhenCondition:(NSInteger)*condition*

Return Value

YES if the lock could be acquired, NO otherwise.

Discussion

As part of its implementation, this method invokes [lockWhenCondition:beforeDate:](#) (page 8). This method returns immediately.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [tryLock](#) (page 9)

Declared In

NSLock.h

unlockWithCondition:

Relinquishes the lock and sets the receiver's condition.

- (void)unlockWithCondition:(NSInteger)*condition*

Parameters

condition

The user-defined condition for the lock. The value of *condition* is user-defined; see the class description for more information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lockWhenCondition:](#) (page 7)

Declared In

NSLock.h

Document Revision History

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

| Date | Notes |
|------------|---|
| 2007-01-15 | Updated for Mac OS X v10.5. |
| 2006-05-23 | First publication of this content as a separate document. |

REVISION HISTORY

Document Revision History

Index

C

condition [instance method 6](#)

I

initWithCondition: [instance method 7](#)

L

lockBeforeDate: [instance method 7](#)

lockWhenCondition: [instance method 7](#)

lockWhenCondition:beforeDate: [instance method 8](#)

N

name [instance method 8](#)

S

setName: [instance method 9](#)

T

tryLock [instance method 9](#)

tryLockWhenCondition: [instance method 10](#)

U

unlockWithCondition: [instance method 10](#)