

NXRecursiveLock

Inherits From: Object
Conforms To: NXLock
Declared In: machkit/NXLock.h

Class Description

An NXRecursiveLock locks a critical section of code such that a single thread can re acquire the lock multiple times without deadlocking, while preventing access by other threads. Note that this implies that a recursive lock will not protect a critical section from a signal handler interrupting the thread holding the lock. Here is an example where a recursive lock functions properly but other lock types would deadlock:

```
// create the lock only once!  
NXRecursiveLock *theLock = [[NXRecursiveLock alloc] init];  
/* ... other code */  
[theLock lock];  
/* ... possibly a long time of fussing with global data... */  
    [theLock lock]; //possibly invoked in a subroutine  
    [theLock unlock];  
[theLock unlock];
```

The NXConditionLock, NXLock, NXRecursiveLock, and NXSpinLock classes all implement the NXLock protocol with various features and performance characteristics; see the other class descriptions for more information.

Instance Variables

None declared in this class.

Method Types

Acquire or release a lock – lock
 – unlock

Instance Methods

lock

– **lock**

Waits until the lock isn't in use by another thread, then grabs the lock and increments an internal counter indicating how many times the lock is held by the current thread.

unlock

– **unlock**

Decrements the internal count indicating how many times the lock is held by the current thread. If the lock is no longer in use by the thread, it is released for use by the next requestor.