



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'APR::ThreadRWLock.3pm'

\$ man APR::ThreadRWLock.3pm

libapache2-mod-perl2-2.0.12::doUserlibapache2-mod-perl2-2.0.12::docs::api::APR::ThreadRWLock(3pm)

NAME

APR::ThreadRWLock - Perl API for APR thread read/write locks

Synopsis

```
use APR::ThreadRWLock ();

my $mutex = APR::ThreadRWLock->new($r->pool);

$mutex->rdlock;
$mutex->wrlock;
$mutex->tryrdlock;
$mutex->trywrlock;
$mutex->unlock;
```

Description

"APR::ThreadRWLock" interfaces APR thread read/write locks.

See `srclib/apr/locks/unix/thread_rwlock.c` in your Apache source tree. At the time of this writing these methods are not supported on all platforms. Thus, check your libraries!

API

"APR::ThreadRWLock" provides the following functions and/or methods:

Unsupported API

"APR::ThreadRWLock" also provides auto-generated Perl interface for a few other methods which aren't tested at the moment and therefore their API is a subject to change. These methods will be finalized later as a need arises. If you want to rely on any of the following methods please contact the the mod_perl development mailing list so we can help each other take the steps necessary to shift the method to an officially supported API.

"DESTROY"

META: Autogenerated - needs to be reviewed/completed

Destroy the lock and free the associated memory.

```
$lock->DESTROY();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock to destroy.

ret: no return value

since: subject to change

"rdlock"

META: Autogenerated - needs to be reviewed/completed

Acquire the read lock for the given lock. The calling thread acquires the read lock if a writer does not hold the lock and there are no writers blocked on the lock. Otherwise it is put to sleep until these conditions are met.

```
$ret = $lock->rdlock();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock on which to acquire the lock.

ret: \$ret (integer)

since: subject to change

"tryrdlock"

META: Autogenerated - needs to be reviewed/completed

Performs the same operation as "rdlock" with the exception that the function shall fail if the thread would be blocked.

```
$ret = $lock->tryrdlock();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock on which to acquire the lock.

ret: \$ret (integer)

since: subject to change

"wrlock"

META: Autogenerated - needs to be reviewed/completed

Acquire the write lock for the given lock. The calling thread acquires the write lock if no other thread (reader or writer) holds it. Otherwise it is put to sleep until this condition is met.

```
$ret = $lock->wrlock();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock on which to acquire the lock.

ret: \$ret (integer)

since: subject to change

"trywrlock"

META: Autogenerated - needs to be reviewed/completed

Performs the same operation as "wrlock" with the exception that the function shall fail if the thread would be blocked.

```
$ret = $lock->trywrlock();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock on which to acquire the lock.

ret: \$ret (integer)

since: subject to change

"new"

Create a new lock

```
my $lock = APR::ThreadRWLock->new($p);
```

obj: "APR::ThreadRWLock" (class name)

arg1: \$p ("APR::Pool object")

ret: \$lock ("APR::ThreadRWLock object")

since: subject to change

"pool_get"

META: Autogenerated - needs to be reviewed/completed

META: should probably be renamed to pool(), like all other pool accessors

Get the pool used by this thread_lock.

```
$ret = $obj->pool_get();
```

obj: \$obj ("APR::ThreadRWLock object")

ret: \$ret ("APR::Pool object")

apr_pool_t the pool

since: subject to change

"unlock"

META: Autogenerated - needs to be reviewed/completed

Release the lock for the given lock.

```
$ret = $lock->unlock();
```

obj: \$lock ("APR::ThreadRWLock object")

the lock from which to release the lock.

ret: \$ret (integer)

since: subject to change

See Also

[mod_perl 2.0 documentation.](#)

Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

Authors

The mod_perl development team and numerous contributors.

perl v5.34.0

libapache2-mod-perl2-2.0.12::docs::api::APR::ThreadRWLock(3pm)