

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

## 1.1 Synopsis

```
use APR::ThreadRWLock ();

my $mutex = APR::ThreadRWLock->new($r->pool);
$mutex->rdlock;
$mutex->wrlock;
$mutex->tryrdlock;
$mutex->trywrlock;
$mutex->unlock;
```

## 1.2 Description

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

See *src/lib/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!

## 1.3 API

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

## 1.4 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.

### 1.4.1 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

## 1.4.2 *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**

## 1.4.3 *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**

## 1.4.4 *wrlock*

META: Autogenerated - needs to be reviewed/completed

Acquire the write lock for the given lock. The calling thread acquires the write lock if 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**

## 1.4.5 *trywlock*

META: Autogenerated - needs to be reviewed/completed

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

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

- **obj:** `$lock (APR::ThreadRWLock object)`

the lock on which to acquire the lock.

- **ret:** `$ret (integer)`
- **since:** subject to change

## 1.4.6 *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

## 1.4.7 *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

## 1.4.8 *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**

## 1.5 See Also

mod\_perl 2.0 documentation.

## 1.6 Copyright

mod\_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

## 1.7 Authors

The mod\_perl development team and numerous contributors.



## Table of Contents:

1	APR::ThreadRWLock - Perl API for APR thread read/write locks	1
1.1	Synopsis	2
1.2	Description	2
1.3	API	2
1.4	Unsupported API	2
1.4.1	DESTROY	2
1.4.2	rdlock	3
1.4.3	tryrdlock	3
1.4.4	wrlock	3
1.4.5	trywrlock	4
1.4.6	new	4
1.4.7	pool_get	4
1.4.8	unlock	5
1.5	See Also	5
1.6	Copyright	5
1.7	Authors	5