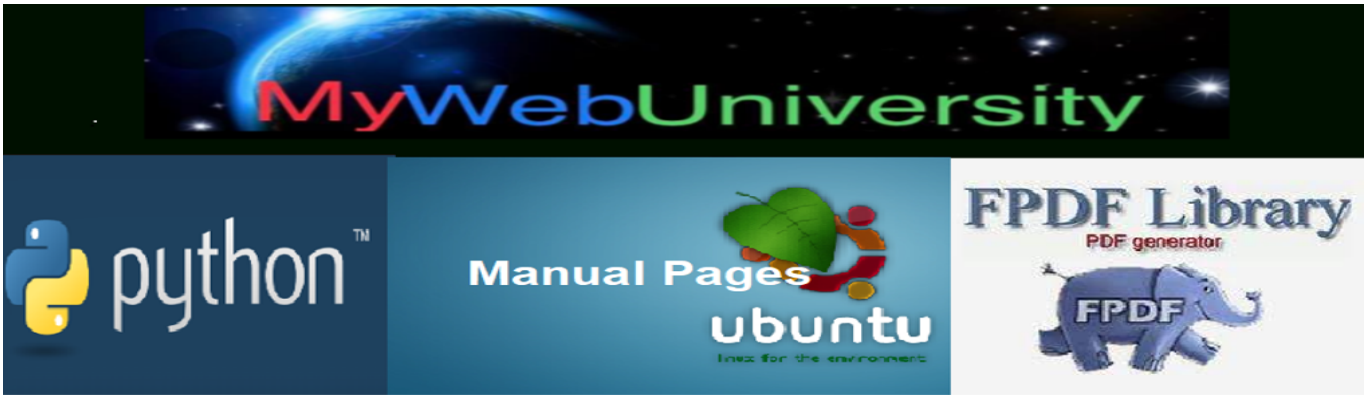

Rocky Enterprise Linux 9.2 Manual Pages on command 'APR::ThreadMutex.3pm'

$ man APR::ThreadMutex.3pm

libapache2-mod-perl2-2.0.12::doUseralibapache2-mod-perl2-2.0.12::docs::api::APR::ThreadMutex(3pm)

NAME

APR::ThreadMutex - Perl API for APR thread mutexes

Synopsis

```
use APR::ThreadMutex ();

my $mutex = APR::ThreadMutex->new($r->pool);
$mutex->lock;
$mutex->unlock;
$mutex->trylock;
```

Description

"APR::ThreadMutex" interfaces APR thread mutexes.

API

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

Unsupported API

"APR::ThreadMutex" 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 mutex and free the memory associated with the lock.

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

obj: $mutex ("APR::ThreadMutex object")

the mutex to destroy.

ret: no return value

since: subject to change

"lock"

META: Autogenerated - needs to be reviewed/completed

Acquire the lock for the given mutex. If the mutex is already locked, the current thread will be put to sleep until the lock becomes available.

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

obj: $mutex ("APR::ThreadMutex object")

the mutex on which to acquire the lock.

ret: $ret (integer)

since: subject to change

"new"

Create a new mutex

```
  my $mutex = APR::ThreadMutex->new($p);
```

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

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

ret: $mutex ("APR::ThreadMutex 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_mutex.

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

obj: $obj ("APR::ThreadMutex object")

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

apr_pool_t the pool

since: subject to change

"trylock"

META: Autogenerated - needs to be reviewed/completed

Attempt to acquire the lock for the given mutex. If the mutex has already been acquired,
the call returns immediately with APR_EBUSY. Note: it is important that the
APR_STATUS_IS_EBUSY(s) macro be used to determine if the return value was APR_EBUSY, for
portability reasons.

```
  $ret = $mutex->trylock();
```

obj: $mutex ("APR::ThreadMutex object")

the mutex on which to attempt the lock acquiring.

ret: $ret (integer)

since: subject to change

"unlock"

META: Autogenerated - needs to be reviewed/completed

Release the lock for the given mutex.

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

obj: $mutex ("APR::ThreadMutex object")

the mutex from which to release the lock.

ret: $ret (integer)

since: subject to change

See Also

mod_perl 2.0 documentation.

Copyright

Authors

The mod_perl development team and numerous contributors.