

**NAME**

"Test::Future" – unit test assertions for Future instances

**SYNOPSIS**

```
use Test::More tests => 2;
use Test::Future;

no_pending_futures {
    my $f = some_function();

    is( $f->get, "result", 'Result of the some_function()' );
} 'some_function() leaves no pending Futures';
```

**DESCRIPTION**

This module provides unit testing assertions that may be useful when testing code based on, or using Future instances or subclasses.

**FUNCTIONS****no\_pending\_futures**

```
no_pending_futures( \&code, $name )
```

*Since version 0.29.*

Runs the given block of code, while keeping track of every Future instance constructed while doing so. After the code has returned, each of these instances are inspected to check that they are not still pending. If they are all either ready (by success or failure) or cancelled, the test will pass. If any are still pending then the test fails.

If Devel::MAT is installed, it will be used to write a memory state dump after a failure. It will create a *.pmat* file named the same as the unit test, but with the trailing *.t* suffix replaced with *-TEST.pmat* where TEST is the number of the test that failed (in case there was more than one). A list of addresses of Future instances that are still pending is also printed to assist in debugging the issue.

It is not an error if the code does not construct any Future instances at all. The block of code may contain other testing assertions; they will be run before the assertion by `no_pending_futures` itself.

**AUTHOR**

Paul Evans <leonerd@leonerd.org.uk>