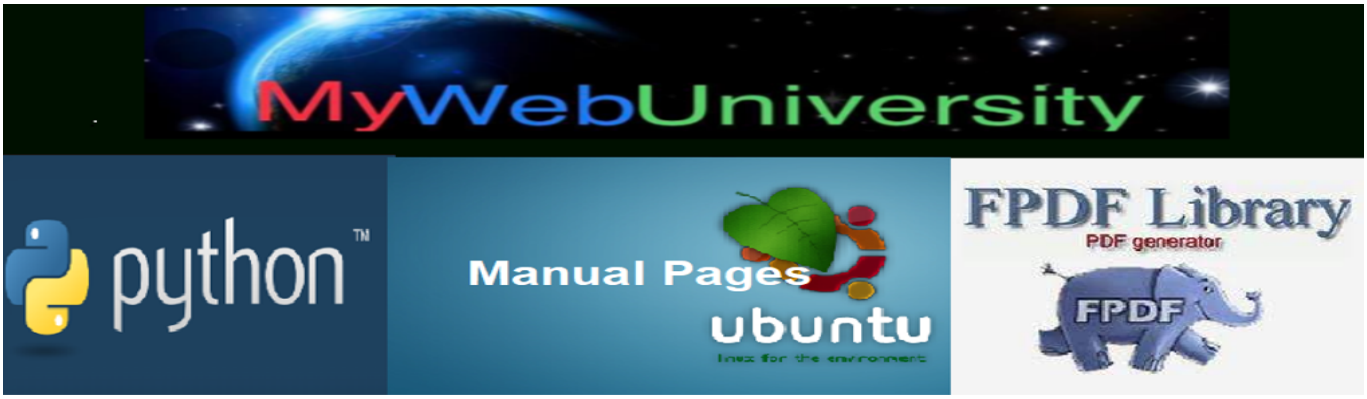

Rocky Enterprise Linux 9.2 Manual Pages on command 'Apache::TestMB.3pm'

$ man Apache::TestMB.3pm

Apache::TestMB(3pm) User Contributed Perl Documentation Apache::TestMB(3pm)

NAME

Apache::TestMB - Subclass of Module::Build to support Apache::Test

SYNOPSIS

Standard process for building & installing modules:

```
perl Build.PL
./Build
./Build test
./Build install
```

Or, if you're on a platform (like DOS or Windows) that doesn't like the "./" notation, you can do this:

```
perl Build.PL
perl Build
perl Build test
perl Build install
```

DESCRIPTION

This class subclasses "Module::Build" to add support for testing Apache integration with

Apache::Test. It is broadly based on "Apache::TestMM", and as such adds a number of build actions to a the Build script, while simplifying the process of creating Build.PL scripts.

Here's how to use "Apache::TestMB" in a Build.PL script:

```
use Module::Build;

my $build_pkg = eval { require Apache::TestMB }
    ? 'Apache::TestMB' : 'Module::Build';

my $build = $build_pkg->new(
    module_name => 'My::Module',
);
$build->create_build_script;
```

This is identical to how "Module::Build" is used. Not all target systems may have "Apache::Test" (and therefore "Apache::TestMB" installed, so we test for it to be installed, first. But otherwise, its use can be exactly the same. Consult the Module::Build documentation for more information on how to use it; Module::Build::Cookbook may be especially useful for those looking to migrate from "ExtUtils::MakeMaker".

INTERFACE

Build

With the above script, users can build your module in the usual "Module::Build" way:

```
perl Build.PL
./Build
./Build test
./Build install
```

If "Apache::TestMB" is installed, then Apache will be started before tests are run by the "test" action, and shut down when the tests complete. Note that "Build.PL" can be called "Apache::Test"-specific options in addition to the usual "Module::Build" options. For

example:

```
perl Build.PL -apxs /usr/local/apache/bin/apxs
```

Consult the Apache::Test documentation for a complete list of options.

In addition to the actions provided by "Module::Build" ("build", "clean", "code", "test", etc.), "Apache::TestMB" adds a few extra actions:

test_clean

This action cleans out the files generated by the test script, t/TEST. It is also executed by the "clean" action.

run_tests

This action actually the tests by executing the test script, t/TEST. It is executed by the "test" action, so most of the time it won't be executed directly.

testcover

"Apache::TestMB" overrides this action from "Module::Build" in order to prevent the "Apache::Test" preference files from being included in the test coverage.

Constructor

new

The "new()" constructor takes all the same arguments as its parent in "Module::Build", but can optionally accept one other parameter:

apache_test_script

The name of the "Apache::Test" test script. The default value is t/TEST, which will work in the vast majority of cases. If you wish to specify your own file name, do so with a relative file name using Unix-style paths; the file name will automatically be converted for the local platform.

When "new()" is called it does the following:

? Processes the "Apache::Test"-specific options in @ARGV. See the Apache::Test documentation for a complete list of options.

? Sets the name of the "Apache::Test" test script to t/TEST, unless it was explicitly specified by the "apache_test_script" parameter.

? Calls "generate_script()" to generate "Apache::Test" test script, usually t/TEST.

Instance Methods

apache_test_args

Returns a hash reference containing all of the settings specified by options passed to Build.PL, or explicitly added to @ARGV in Build.PL. Consult the Apache::Test documentation for a complete list of options.

apache_test_script

Gets or sets the file name of the "Apache::Test" test script.

generate_script

```
$build->generate_script;
$build->generate_script('t/FOO');
$build->generate_script(undef, 'Apache::TestRun');
```

This method is called by "new()", so in most cases it can be ignored. If you'd like it to use other than the default arguments, you can call it explicitly in Build.PL and pass it the arguments you desire. It takes two optional arguments:

? The name of the "Apache::Test" test script. Defaults to the value returned by "apache_test_script()".

? The name of an "Apache::Test" test running class. Defaults to "Apache::TestRunPerl".

If there is an existing t/TEST.PL (or a script with the same name as specified by the "apache_test_script" parameter but with .PL appended to it), then that script will be used as the template for the test script. Otherwise, a simple test script will be written similar to what would be written by "Apache::TestRun::generate_script()" (although that function is not aware of the arguments passed to Build.PL, so use this one instead!).

SEE ALSO

Apache::TestRequest

Demonstrates how to write tests to send requests to the Apache server run by "./Build test".

Module::Build

The parent class for "Apache::TestMB"; consult it's documentation for more on its interface.

<http://www.perl.com/pub/a/2003/05/22/testing.html>

This article by Geoffrey Young explains how to configure Apache and write tests for your module using Apache::Test. Just use "Apache::TestMB" instead of "Apache::TestMM" to update it for use with "Module::Build".

AUTHOR

David Wheeler

Questions can be asked at the test-dev <at> httpd.apache.org list. For more information see: http://httpd.apache.org/test/ and http://perl.apache.org/docs/general/testing/testing.html.