



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 'Apache::TestMM.3pm'

\$ man Apache::TestMM.3pm

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

NAME

Apache::TestMM - Provide MakeMaker Wrapper Methods

SYNOPSIS

```
require Apache::TestMM;

# import MY::test and MY::clean overrides for MM

Apache::TestMM->import(qw(test clean));

# parse command line args

Apache::TestMM::filter_args();

# autogenerate the script

Apache::TestMM::generate_script('t/TEST');
```

DESCRIPTION

"Apache::TestMM" provides wrappers for the "ExtUtils::MakeMaker" craft, making it easier to extend the autogenerated Makefile with "Apache::Test".

FUNCTIONS

"import"

```
use Apache::TestMM qw(test clean);
```

or:

```
Apache::TestMM->import(qw(test clean));
```

Imports "MY::" overrides for the default "ExtUtils::MakeMaker" test and clean targets, as

if you have defined:

```
sub MY::test {...}
```

```
sub MY::clean {...}
```

in Makefile.PL. "Apache::TestMM" does this for you so that these Makefile targets will run the Apache server and the tests for it, and clean up after its mess.

"filter_args"

```
push @ARGV, '-apxs', $apxs_path;
```

```
Apache::TestMM::filter_args();
```

```
WriteMakefile(...);
```

When "WriteMakefile()" is called it parses @ARGV, hoping to find special options like "PREFIX=/home/stas/perl". "Apache::Test" accepts a lot of configuration options of its own. When "Apache::TestMM::filter_args()" is called, it removes any "Apache::Test"-specific options from @ARGV and stores them internally, so when "WriteMakefile()" is called they aren't in @ARGV and thus won't be processed by "WriteMakefile()".

The options can be set when Makefile.PL is called:

```
% perl Makefile.PL -apxs /path/to/apxs
```

Or you can push them manually to @ARGV from the code:

```
push @ARGV, '-apxs', $apxs_path;
```

When:

```
Apache::TestMM::generate_script('t/TEST');
```

is called, "Apache::Test"-specific options extracted by "Apache::TestMM::filter_args()" are written to the autogenerated file. In our example, the autogenerated t/TEST will include:

```
%Apache::TestConfig::Argv = qw(apxs /path/to/apxs);
```

which is going to be used by the "Apache::Test" runtime.

The other frequently used options are: "-httpd", telling where to find the httpd (usually when the "-apxs" option is not used), "-libmodperl" to use a specific mod_perl shared object (if your mod_perl is built as DSO), "-maxclients" to change the default number of the configured "MaxClients" directive, "-port" to start the server on a specific port, etc. To get the complete list of available configuration options and their purpose and syntax, run:

```
% perl -MApache::TestConfig -le 'Apache::TestConfig::usage()'
```

You may wish to document some of these in your application's README file, especially the "-apxs" and "-httpd" options.

"generate_script"

```
Apache::TestMM::generate_script('t/TEST');
```

"generate_script()" accepts the name of the script to generate and will look for a template with the same name and suffix .PL. So in our example it'll look for t/TEST.PL. The autogenerated script t/TEST will include the contents of t/TEST.PL, and special directives, including any configuration options passed via "filter_args()" called from Makefile.PL, special fixup code, etc.

perl v5.34.0

2022-02-06

Apache::TestMM(3pm)