



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::TestRun.3pm'

\$ man Apache::TestRun.3pm

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

NAME

Apache::TestRun - Run the test suite

SYNOPSIS

DESCRIPTION

The "Apache::TestRun" package controls the configuration and running of the test suite.

METHODS

Several methods are sub-classable, if the default behavior should be changed.

"bug_report"

The "bug_report()" method is executed when "t/TEST" was executed with the "-bugreport" option, and "make test" (or "t/TEST") fail. Normally this is callback which you can use to tell the user how to deal with the problem, e.g. suggesting to read some document or email some details to someone who can take care of it. By default nothing is executed.

The "-bugreport" option is needed so this feature won't become annoying to developers themselves. It's automatically added to the "run_tests" target in Makefile. So if you repeatedly have to test your code, just don't use "make test" but run "t/TEST" directly.

Here is an example of a custom "t/TEST"

```
My::TestRun->new->run(@ARGV);
```

```
package My::TestRun;
```

```
use base 'Apache::TestRun';
```

```
sub bug_report {
```

```
    my $self = shift;
```

```
    print <<EOI;
```

```
    +-----+
```

```
    | Please file a bug report: http://perl.apache.org/bugs/ |
```

```
    +-----+
```

```
    EOI
```

```
}
```

"pre_configure"

The "pre_configure()" method is executed before the configuration for "Apache::Test" is generated. So if you need to adjust the setup before httpd.conf and other files are autogenerated, this is the right place to do so.

For example if you don't want to inherit a LoadModule directive for mod_apreq.so but to make sure that the local version is used, you can sub-class "Apache::TestRun" and override this method in t/TEST.PL:

```
package My::TestRun;
```

```
use base 'Apache::TestRun';
```

```
use Apache::TestConfig;
```

```
__PACKAGE__->new->run(@ARGV);
```

```
sub pre_configure {
```

```
    my $self = shift;
```

```
    # Don't load an installed mod_apreq
```

```
    Apache::TestConfig::autoconfig_skip_module_add('mod_apreq.c');
```

```
$self->SUPER::pre_configure();  
}
```

Notice that the extension is .c, and not .so.

Don't forget to run the super class' c<pre_configure(> method.

"new_test_config"

META: to be completed

perl v5.34.0

2022-02-06

Apache::TestRun(3pm)