

**NAME**

MooseX::SimpleConfig – A Moose role for setting attributes from a simple configuration file

**VERSION**

version 0.11

**SYNOPSIS**

```
## A YAML configfile named /etc/my_app.yaml:
foo: bar
baz: 123

## In your class
package My::App;
use Moose;

with 'MooseX::SimpleConfig';

has 'foo' => (is => 'ro', isa => 'Str', required => 1);
has 'baz' => (is => 'rw', isa => 'Int', required => 1);

# ... rest of the class here

## in your script
#!/usr/bin/perl

use My::App;

my $app = My::App->new_with_config(configfile => '/etc/my_app.yaml');
# ... rest of the script here

#####
##### combined with MooseX::Getopt:

## In your class
package My::App;
use Moose;

with 'MooseX::SimpleConfig';
with 'MooseX::Getopt';

has 'foo' => (is => 'ro', isa => 'Str', required => 1);
has 'baz' => (is => 'rw', isa => 'Int', required => 1);

# ... rest of the class here

## in your script
#!/usr/bin/perl

use My::App;

my $app = My::App->new_with_options();
# ... rest of the script here

## on the command line
% perl my_app_script.pl -configfile /etc/my_app.yaml -otherthing 123
```

**DESCRIPTION**

This role loads simple files to set object attributes. It is based on the abstract role MooseX::ConfigFromFile, and uses Config::Any to load your configuration file. Config::Any will in turn support any of a variety of different config formats, detected by the file extension. See



Config::Any for more details about supported formats.

To pass additional arguments to Config::Any you must provide a `config_any_args()` method, for example:

```
sub config_any_args {
    return {
        driver_args => { General => { '-InterPolateVars' => 1 } }
    };
}
```

Like all MooseX::ConfigFromFile -derived file loaders, this module is automatically supported by the MooseX::Getopt role as well, which allows specifying `-configfile` on the command line.

## ATTRIBUTES

### configfile

Provided by the base role MooseX::ConfigFromFile. You can provide a default configuration file pathname like so:

```
has '+configfile' => ( default => '/etc/myapp.yaml' );
```

You can pass an array of filenames if you want, but as usual the array has to be wrapped in a sub ref.

```
has '+configfile' => ( default => sub { [ '/etc/myapp.yaml', '/etc/myapp_local.' ] } );
```

Config files are trivially merged at the top level, with the right-hand files taking precedence.

## CLASS METHODS

### new\_with\_config

Provided by the base role MooseX::ConfigFromFile. Acts just like regular `new()`, but also accepts an argument `configfile` to specify the file from which to load other attributes. Explicit arguments to `new_with_config` will override anything loaded from the file.

### get\_config\_from\_file

Called internally by either `new_with_config` or MooseX::Getopt's `new_with_options`. Invokes Config::Any to parse configfile.

## AUTHOR

Brandon L. Black <blblack AT gmail DOT com>

## COPYRIGHT AND LICENSE

This software is copyright (c) 2007 by Brandon L. Black <blblack AT gmail DOT com>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

## CONTRIBUTORS

- Karen Etheridge <ether AT cpan DOT org>
- Tomas Doran <bobtfish AT bobtfish DOT net>
- Brandon L Black <blblack AT gmail DOT com>
- Alexander Hartmaier <alex DOT hartmaier AT gmail DOT com>
- lestrrat <lestrrat+github AT gmail DOT com>
- XXXXXX XXXXXXXX <sromanov AT cpan DOT org>
- Yuval Kogman <nothingmuch AT woobling DOT org>
- Zbigniew Lukasiak <zby AT cpan DOT org>
- Alex Howarth <alex DOT howarth AT gmail DOT com>

