

NAME

MouseX::ConfigFromFile – An abstract Mouse role for setting attributes from a configfile

SYNOPSIS

A real role based on this abstract role:

```

package MyApp::ConfigRole;
use Mouse::Role;
with 'MouseX::ConfigFromFile';

use MyApp::ConfigLoader;

sub get_config_from_file {
    my ($class, $file) = @_;

    my $config_hashref = MyApp::ConfigLoader->load($file);

    return $config_hashref;
}

```

A class that uses it:

```

package MyApp;
use Mouse;
with 'MyApp::ConfigRole';

# optionally, default the configfile:
has '+configfile' => ( default => '/tmp/myapp.yml' );

```

A script that uses the class with a configfile:

```

my $app = MyApp->new_with_config(
    configfile => '/etc/myapp.yml',
    other_opt  => 'foo',
);

```

DESCRIPTION

This is an abstract role which provides an alternate constructor for creating objects using parameters passed in from a configuration file. The actual implementation of reading the configuration file is left to concrete subroles.

It declares an attribute `configfile` and a class method `new_with_config`, and requires that concrete roles derived from it implement the class method `get_config_from_file`.

Attributes specified directly as arguments to `new_with_config` supercede those in the configfile.

METHODS

new_with_config(%params?)

This is an alternate constructor, which knows to look for the `configfile` option in its arguments and use that to set attributes. It is much like `MouseX::Getopts`' `new_with_options`.

Example:

```

my $app = MyApp->new_with_config( configfile => '/etc/foo.yaml' );

```

Explicit arguments will override anything set by the configfile.

get_config_from_file(\$file)

This method is not implemented in this role, but it is required of all subroles. Its two arguments are the class name and the configfile, and it is expected to return a hashref of arguments to pass to `new()` which are sourced from the configfile.

Example:

```

sub get_config_from_file {
    my ($class, $file) = @_;

    my $config = {};
}

```



```
        # ... load config from $file ...

        return $config;
    }
}
```

PROPERTIES

configfile

This is a Path::Class::File object which can be coerced from a regular path name string. This is the file your attributes are loaded from. You can add a default configfile in the class using the role and it will be honored at the appropriate time:

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

AUTHOR

NAKAGAWA Masaki <masaki AT cpan DOT org>

THANKS TO

Brandon L. Black, “AUTHOR” in MooseX::ConfigFromFile

LICENSE

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

SEE ALSO

Mouse, Mouse::Role, MooseX::Types::Path::Class, MooseX::ConfigFromFile

