

MooseX::Declare::Syntax::MooseSetup(3pm) [Use MooseX::Declare::Syntax::MooseSetup\(3pm\)](#) [Upstream Contributed Perl Documentation](#)

NAME

MooseX::Declare::Syntax::MooseSetup – Common Moose namespaces declarations

VERSION

version 0.43

DESCRIPTION

This role is basically an extension to NamespaceHandling. It adds all the common parts for Moose namespace definitions. Examples of this role can be found in the class and role keywords.

METHODS

auto_make_immutable

Bool Object->auto_make_immutable ()

Since Moose::Roles can't be made immutable (this is not a bug or a missing feature, it would make no sense), this always returns false.

imported_moose_symbols

List Object->imported_moose_symbols ()

This will return confess and blessed by default to provide as additional imports to the namespace.

import_symbols_from

Str Object->import_symbols_from ()

The namespace from which the additional imports will be imported. This will return Moose by default.

CONSUMES

- MooseX::Declare::Syntax::NamespaceHandling
- MooseX::Declare::Syntax::EmptyBlockIfMissing

MODIFIED METHODS

default_inner

ArrayRef default_inner ()

This will provide the following default inner-handlers to the namespace:

- method

A simple Method handler.
- around

This is a MethodModifier handler that will start the signature of the generated method with \$orig: \$self to provide the original method in \$orig.
- after
- before
- override
- augment

These four handlers are MethodModifier instances.
- clean

This is an instance of the Clean keyword handler.

The original method will never be called and all arguments are ignored at the moment.

setup_inner_for

Object->setup_inner_for (ClassName \$class)

This will install a with function that will push its arguments onto a global storage array holding the roles of the current namespace.

add_namespace_customizations

Object->add_namespace_customizations (Object \$context, Str \$package, HashRef \$options)

After all other customizations, this will first add code to import the “imported_moose_symbols” from the package returned in “import_symbols_from” to the preamble.

Then it will add a code part that will immutabilize the class to the cleanup code if the



MooseX::Declare::Syntax::MooseSetup(3pm) [Use\(3pm\)](#) [Contributed Perl Document](#) [MooseX::Declare::Syntax::MooseSetup\(3pm\)](#)

“auto_make_immutable” method returned a true value and `$options->{is}{mutable}` does not exist.

handle_post_parsing

`CodeRef Object->handle_post_parsing (Object $context, Str $package, Str|Object :`

Generates a callback that sets up the roles in the global role storage for the current namespace. The `$name` parameter will be the specified name (in contrast to `$package` which will always be the fully qualified name) or the anonymous metaclass instance if none was specified.

SEE ALSO

- `MooseX::Declare`
- `Moose`

AUTHOR

Florian Ragwitz <rafl AT debian DOT org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2008 by Florian Ragwitz.

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

