NAME

MooseX::XSAccessor − use Class::XSAccessor to speed up Moose accessors

SYNOPSIS

```
package MyClass;

use Moose;
use MooseX::XSAccessor;

has foo => (...);
```

DESCRIPTION

This module accelerates Moose−generated accessor, reader, writer and predicate methods using Class::XSAccessor. You get a speed-up for no extra effort. It is automatically applied to every attribute in the class.

The use of the following features of Moose attributes prevents a reader from being accelerated:

- Lazy builder or lazy default.
- Auto-deref. (Does anybody use this anyway??)

The use of the following features prevents a writer from being accelerated:

- Type constraints (except `Any`; `Any` is effectively a no-op).
- Triggers
- Weak references

An `rw` accessor is effectively a reader and a writer glued together, so both of the above lists apply.

Predicates can always be accelerated, provided you're using Class::XSAccessor 1.17 or above.

Clearers can not be accelerated (as of current versions of Class::XSAccessor).

Functions

This module also provides one function, which is not exported so needs to be called by its full name.

`MooseX::XSAccessor::is_xs($sub)`

Returns a boolean indicating whether a sub is an XSUB.

`$sub` may be a coderef, Class::MOP::Method object, or a qualified sub name as a string (e.g. `"MyClass::foo"`).

Chained accessors and writers

MooseX::XSAccessor can detect chained accessors and writers created using MooseX::Attribute::Chained, and can accelerate those too.

```
package Local::Class;
use Moose;
use MooseX::XSAccessor;
use MooseX::Attribute::Chained;

has foo => (traits => ["Chained"], is => "rw");
has bar => (traits => ["Chained"], is => "ro", writer => "_set_bar");
has baz => (                       is => "rw");  # not chained

my $obj = "Local::Class"->new;
$obj->foo(1)->_set_bar(2);
print $obj->dump;
```

Lvalue accessors

MooseX::XSAccessor will detect lvalue accessors created with MooseX::LvalueAttribute and, by default, skip accelerating them.

However, by setting `$MooseX::XSAccessor::LVALUE` to true (preferably using the `local` Perl keyword), you can force it to accelerate those too. This introduces a visible change in behaviour though. MooseX::LvalueAttribute accessors normally allow two patterns for setting the value:

```
$obj->foo = 42;   # as an lvalue
$obj->foo(42);    # as a method call
```

However, once accelerated, they may *only* be set as an lvalue. For this reason, setting `$MooseX::XSAccessor::LVALUE` to true is considered an experimental feature.

HINTS

- Make attributes read-only when possible. This means that type constraints and coercions will only apply to the constructor, not the accessors, enabling the accessors to be accelerated.
- If you do need a read-write attribute, consider making the main accessor read-only, and having a separate writer method. (Like MooseX::SemiAffordanceAccessor.)
- Make defaults eager instead of lazy when possible, allowing your readers to be accelerated.
- If you need to accelerate just a specific attribute, apply the attribute trait directly:

```
package MyClass;

use Moose;

has foo => (
   traits => ["MooseX::XSAccessor::Trait::Attribute"],
   ...,
);
```

- If you don't want to add a dependency on MooseX::XSAccessor, but do want to use it if it's available, the following code will use it optionally:

```
package MyClass;

use Moose;
BEGIN { eval "use MooseX::XSAccessor" };

has foo => (...);
```

CAVEATS

- Calling a writer method without a parameter in Moose does not raise an exception:

```
$person->set_name();    # sets name attribute to "undef"
```

  However, this is a fatal error in Class::XSAccessor.

- MooseX::XSAccessor does not play nice with attribute traits that alter accessor behaviour, or define additional accessors for attributes. MooseX::SetOnce is an example thereof. MooseX::Attribute::Chained is handled as a special case.
- MooseX::XSAccessor only works on blessed hash storage; not e.g. MooseX::ArrayRef or MooseX::InsideOut. MooseX::XSAccessor is usually able to detect such situations and silently switch itself off.

BUGS

Please report any bugs to <http://rt.cpan.org/Dist/Display.html?Queue=MooseX-XSAccessor>.

SEE ALSO

MooseX::XSAccessor::Trait::Attribute.

Moose, Moo, Class::XSAccessor.

AUTHOR

Toby Inkster <tobyink AT cpan DOT org>.

COPYRIGHT AND LICENCE

DISCLAIMER OF WARRANTIES

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.