

NAME

MooseX::Types::Perl – Moose types that check against Perl syntax

VERSION

version 0.101343

SYNOPSIS

```
use MooseX::Types::Perl qw(
    DistName

    ModuleName
    PackageName

    Identifier
    SafeIdentifier

    LaxVersionStr
    StrictVersionStr
    VersionObject
);
```

DESCRIPTION

This library provides Moose types for checking things (mostly strings) against syntax that is, or is a reasonable subset of, Perl syntax.

TYPES**ModuleName****PackageName**

These types are identical, and expect a string that could be a package or module name. That's basically a bunch of identifiers stuck together with double-colons. One key quirk is that parts of the package name after the first may begin with digits.

The use of an apostrophe as a package separator is not permitted.

DistName

The DistName type checks for a string like `MooseX-Types-Perl`, the sort of thing used to name CPAN distributions. In general, it's like the more familiar `ModuleName`, but with hyphens instead of double-colons.

In reality, a few distribution names may not match this pattern — most famously, `CGI.pm` is the name of the distribution that contains `CGI`. These exceptions are few and far between, and deciding what a `LaxDistName` type would look like has not seemed worth it, yet.

Identifier

An Identifier is something that could be used as a symbol name or other identifier (filehandle, directory handle, subroutine name, format name, or label). It's what you put after the sigil (dollar sign, at sign, percent sign) in a variable name. Generally, it's a bunch of alphanumeric characters not starting with a digit.

Although Perl identifiers may contain non-ASCII characters in some circumstances, this type does not allow it. A `UnicodeIdentifier` type may be added in the future.

SafeIdentifier

SafeIdentifiers are just like Identifiers, but omit the single-letter variables `underscore`, `a`, and `b`, as these have special significance.

LaxVersionStr**StrictVersionStr**

Lax and strict version strings use the `is_lax` and `is_strict` methods from `version` to check if the given string would be a valid lax or strict version. `version::Internals` covers the details but basically: lax versions are everything you may do, and strict omit many of the usages best avoided.

VersionObject

Just for good measure, this type is included to check if a value is a version object. Coercions from `LaxVersionStr` (and thus `StrictVersionStr`) are provided.



AUTHOR

Ricardo SIGNES <rjbs AT cpan DOT org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2014 by Ricardo SIGNES.

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

