NAME

CPAN::Meta::Converter - Convert CPAN distribution metadata structures

VERSION

version 2.140640

SYNOPSIS

```
my $struct = decode_json_file('META.json');

my $cmc = CPAN::Meta::Converter->new( $struct );

my $new_struct = $cmc->convert( version => "2" );
```

DESCRIPTION

This module converts CPAN Meta structures from one form to another. The primary use is to convert older structures to the most modern version of the specification, but other transformations may be implemented in the future as needed. (E.g. stripping all custom fields or stripping all optional fields.)

METHODS

new

```
my $cmc = CPAN::Meta::Converter->new( $struct );
```

The constructor should be passed a valid metadata structure but invalid structures are accepted. If no meta-spec version is provided, version 1.0 will be assumed.

convert

```
my $new_struct = $cmc->convert( version => "2" );
```

Returns a new hash reference with the metadata converted to a different form. `convert` will die if any conversion/standardization still results in an invalid structure.

Valid parameters include:

- `version` — Indicates the desired specification version (e.g. "1.0", "1.1" ... "1.4", "2"). Defaults to the latest version of the CPAN Meta Spec.

Conversion proceeds through each version in turn. For example, a version 1.2 structure might be converted to 1.3 then 1.4 then finally to version 2. The conversion process attempts to clean-up simple errors and standardize data. For example, if `author` is given as a scalar, it will converted to an array reference containing the item. (Converting a structure to its own version will also clean-up and standardize.)

When data are cleaned and standardized, missing or invalid fields will be replaced with sensible defaults when possible. This may be lossy or imprecise. For example, some badly structured META.yml files on CPAN have prerequisite modules listed as both keys and values:

```
requires => { 'Foo::Bar' => 'Bam::Baz' }
```

These would be split and each converted to a prerequisite with a minimum version of zero.

When some mandatory fields are missing or invalid, the conversion will attempt to provide a sensible default or will fill them with a value of 'unknown'. For example a missing or unrecognized `license` field will result in a `license` field of 'unknown'. Fields that may get an 'unknown' include:

- abstract
- author
- license

BUGS

Please report any bugs or feature using the CPAN Request Tracker. Bugs can be submitted through the web interface at <http://rt.cpan.org/Dist/Display.html?Queue=CPAN-Meta>

When submitting a bug or request, please include a test-file or a patch to an existing test-file that illustrates the bug or desired feature.

AUTHORS

- David Golden <dagolden@cpan.org>

- Ricardo Signes <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by David Golden and 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.