



Dist::Zilla

Dist::Zilla	13
DESCRIPTION	13
ATTRIBUTES	13
name	13
version	13
abstract	13
main_module	13
license	13
authors	13
files	14
root	14
is_trial	14
plugins	14
distmeta	14
prereqs	14
logger	14
METHODS	15
plugin_named	15
plugins_with	15
find_files	15
stash_named	15
SUPPORT	15
AUTHOR	15
Dist::Zilla::Tutorial	16
SYNOPSIS	16
BUILDING YOUR DIST	17
HOW BUILDS GET BUILT	17
THE GLORIOUS FUTURE	17
AUTHOR	18
Dist::Zilla::Plugin::Prereqs	19
VERSION	19
SYNOPSIS	19
DESCRIPTION	19
AUTHOR	20
COPYRIGHT AND LICENSE	20
Dist::Zilla::Plugin::RemovePrereqs	21
VERSION	21
SYNOPSIS	21
AUTHOR	21
COPYRIGHT AND LICENSE	21
Dist::Zilla::Plugin::MinimumVersionTests	22
VERSION	22
SYNOPSIS	22
DESCRIPTION	22
INSTALLATION	22
AUTHOR	22
COPYRIGHT AND LICENSE	22
Dist::Zilla::Plugin::ReportVersions	23
VERSION	23
SYNOPSIS	23
DESCRIPTION	23

INSTALLATION	23
BUGS AND LIMITATIONS	23
AVAILABILITY	23
AUTHOR	23
COPYRIGHT AND LICENSE	23
Dist::Zilla::Plugin::GatherDir::Template	25
VERSION	25
DESCRIPTION	25
AUTHOR	25
COPYRIGHT AND LICENSE	25
Dist::Zilla::Plugin::GithubMeta	26
VERSION	26
SYNOPSIS	26
DESCRIPTION	26
ATTRIBUTES	26
METHODS	26
SEE ALSO	27
AUTHOR	27
COPYRIGHT AND LICENSE	27
Dist::Zilla::Plugin::InstallGuide	28
VERSION	28
SYNOPSIS	28
DESCRIPTION	28
FUNCTIONS	28
setup_installer	28
INSTALLATION	28
BUGS AND LIMITATIONS	28
AVAILABILITY	28
AUTHOR	28
COPYRIGHT AND LICENSE	29
Dist::Zilla::Plugin::License	30
VERSION	30
DESCRIPTION	30
AUTHOR	30
COPYRIGHT AND LICENSE	30
Dist::Zilla::Plugin::PodVersion	31
VERSION	31
DESCRIPTION	31
AUTHOR	31
COPYRIGHT AND LICENSE	31
Dist::Zilla::Plugin::TemplateModule	32
VERSION	32
DESCRIPTION	32
ATTRIBUTES	32
template	32
AUTHOR	32
COPYRIGHT AND LICENSE	32
Dist::Zilla::Plugin::Prereq	33
VERSION	33
SYNOPSIS	33
AUTHOR	33
COPYRIGHT AND LICENSE	33
Dist::Zilla::Plugin::FinderCode	34
VERSION	34

AUTHOR.....	34
COPYRIGHT AND LICENSE.....	34
Dist::Zilla::Plugin::Git::Init	35
VERSION.....	35
DESCRIPTION	35
AUTHOR.....	35
COPYRIGHT AND LICENSE.....	35
Dist::Zilla::Plugin::PkgVersion	36
VERSION.....	36
SYNOPSIS.....	36
DESCRIPTION	36
AUTHOR.....	36
COPYRIGHT AND LICENSE.....	36
Dist::Zilla::Plugin::Git::Tag.....	37
VERSION.....	37
SYNOPSIS.....	37
DESCRIPTION	37
Plugin options	37
Formatting options.....	37
AUTHOR.....	38
COPYRIGHT AND LICENSE.....	38
Dist::Zilla::Plugin::BumpVersionFromGit	39
VERSION.....	39
SYNOPSIS.....	39
DESCRIPTION	39
SEE ALSO	39
AUTHOR.....	39
COPYRIGHT AND LICENSE.....	39
Dist::Zilla::Plugin::TaskWeaver	41
VERSION.....	41
DESCRIPTION	41
WARNING	41
AUTHOR.....	41
COPYRIGHT AND LICENSE.....	41
Dist::Zilla::Plugin::ModuleInstall.....	43
VERSION.....	43
SYNOPSIS.....	43
DESCRIPTION	43
AUTHOR.....	43
COPYRIGHT AND LICENSE.....	43
Dist::Zilla::Plugin::ChangelogFromGit	44
VERSION.....	44
SYNOPSIS.....	44
DESCRIPTION	44
Subversion and CVS.....	45
AUTHOR.....	45
COPYRIGHT AND LICENSE.....	45
Dist::Zilla::Plugin::PodWeaver.....	46
VERSION.....	46
DESCRIPTION	46
METHODS	46
weaver.....	46
CONFIGURATION.....	46
AUTHOR.....	46

COPYRIGHT AND LICENSE.....	46
Dist::Zilla::Plugin::Git::CommitBuild.....	47
VERSION.....	47
SYNOPSIS.....	47
DESCRIPTION.....	47
AUTHOR.....	47
COPYRIGHT AND LICENSE.....	47
Dist::Zilla::Plugin::PerlTidy.....	48
VERSION.....	48
METHODS.....	48
munged_file.....	48
SYNOPSIS.....	48
DEFAULTS.....	48
SEE ALSO.....	48
AUTHORS.....	48
COPYRIGHT AND LICENSE.....	48
Dist::Zilla::Plugin::MatchManifest.....	49
VERSION.....	49
SYNOPSIS.....	49
DESCRIPTION.....	49
CONFIGURATION AND ENVIRONMENT.....	49
INCOMPATIBILITIES.....	49
BUGS AND LIMITATIONS.....	49
AUTHOR.....	49
COPYRIGHT AND LICENSE.....	50
DISCLAIMER OF WARRANTY.....	50
Dist::Zilla::Plugin::PkgDist.....	51
VERSION.....	51
DESCRIPTION.....	51
AUTHOR.....	51
COPYRIGHT AND LICENSE.....	51
Dist::Zilla::Plugin::TestRelease.....	52
VERSION.....	52
DESCRIPTION.....	52
CREDITS.....	52
AUTHOR.....	52
COPYRIGHT AND LICENSE.....	52
Dist::Zilla::Plugin::MetaProvides.....	53
VERSION.....	53
SYNOPSIS.....	53
DESCRIPTION.....	53
COMPONENT SUMMARY.....	53
::Class.....	53
::Package.....	53
::FromFile.....	54
OPTION SUMMARY.....	54
inherit_version.....	54
values.....	54
inherit_missing.....	54
values.....	54
file.....	55
reader_name.....	55
AUTHOR.....	55
COPYRIGHT AND LICENSE.....	55

Dist::Zilla::Plugin::CheckChangeLog	56
VERSION	56
SYNOPSIS.....	56
DESCRIPTION	56
AUTHOR.....	56
COPYRIGHT AND LICENSE.....	56
Dist::Zilla::Plugin::CompileTests	57
VERSION	57
SYNOPSIS.....	57
DESCRIPTION	57
SEE ALSO	57
AUTHOR.....	58
COPYRIGHT AND LICENSE.....	58
Dist::Zilla::Plugin::ExtraTests	59
VERSION	59
DESCRIPTION	59
AUTHOR.....	59
COPYRIGHT AND LICENSE.....	59
Dist::Zilla::Plugin::NextRelease	60
VERSION	60
SYNOPSIS.....	60
DESCRIPTION	60
AUTHOR.....	61
COPYRIGHT AND LICENSE.....	61
Dist::Zilla::Plugin::UnusedVarsTests	62
VERSION	62
SYNOPSIS.....	62
DESCRIPTION	62
INSTALLATION	62
BUGS AND LIMITATIONS	62
AVAILABILITY	62
AUTHOR.....	62
COPYRIGHT AND LICENSE.....	62
Dist::Zilla::Plugin::Homepage	63
VERSION	63
SYNOPSIS.....	63
DESCRIPTION	63
FUNCTIONS	63
metadata	63
INSTALLATION	63
BUGS AND LIMITATIONS	63
AVAILABILITY	63
AUTHOR.....	63
COPYRIGHT AND LICENSE.....	63
Dist::Zilla::Plugin::PruneFiles	65
VERSION	65
SYNOPSIS.....	65
ATTRIBUTES	65
filenames.....	65
matches	65
AUTHOR.....	65
COPYRIGHT AND LICENSE.....	65
Dist::Zilla::Plugin::PruneCruft	66
VERSION	66

SYNOPSIS.....	66
AUTHOR.....	66
COPYRIGHT AND LICENSE.....	66
Dist::Zilla::Plugin::MetaProvides::Package	67
VERSION.....	67
ROLES.....	67
.....	67
ROLE SATISFYING METHODS	67
provides.....	67
signature: \$plugin->provides()	67
returns: Array of	67
PRIVATE METHODS	67
_packages_for.....	67
signature: \$plugin->_packages_for(\$filename, \$file_content)	67
returns: Array of	67
SEE ALSO	67
AUTHOR.....	67
COPYRIGHT AND LICENSE.....	67
Dist::Zilla::Plugin::ModuleBuild.....	68
VERSION.....	68
DESCRIPTION	68
ATTRIBUTES.....	68
mb_version.....	68
mb_class.....	68
AUTHOR.....	68
COPYRIGHT AND LICENSE.....	68
Dist::Zilla::Plugin::ModuleShareDirs.....	69
VERSION.....	69
SYNOPSIS.....	69
AUTHOR.....	69
COPYRIGHT AND LICENSE.....	69
Dist::Zilla::Plugin::ConfirmRelease	70
VERSION.....	70
DESCRIPTION	70
AUTHOR.....	70
COPYRIGHT AND LICENSE.....	70
Dist::Zilla::Plugin::MetaProvides::Class.....	71
VERSION.....	71
ROLES.....	71
.....	71
ROLE SATISFYING METHODS	71
provides.....	71
signature: \$plugin->provides()	71
returns: Array of	71
PRIVATE METHODS	71
_classes_for.....	71
signature: \$plugin->_classes_for(\$filename, \$file_content)	71
returns: Array of	71
SEE ALSO	71
AUTHOR.....	71
COPYRIGHT AND LICENSE.....	71
Dist::Zilla::Plugin::MetaConfig.....	72
VERSION.....	72
DESCRIPTION	72

AUTHOR.....	72
COPYRIGHT AND LICENSE.....	72
Dist::Zilla::Plugin::Twitter	73
VERSION.....	73
SYNOPSIS.....	73
DESCRIPTION	73
AUTHOR.....	74
COPYRIGHT AND LICENSE.....	74
Dist::Zilla::Plugin::MetaResources.....	75
VERSION.....	75
DESCRIPTION	75
AUTHOR.....	75
COPYRIGHT AND LICENSE.....	75
Dist::Zilla::Plugin::DistINI	76
VERSION.....	76
DESCRIPTION	76
ATTRIBUTES	76
append_file.....	76
AUTHOR.....	76
COPYRIGHT AND LICENSE.....	77
Dist::Zilla::Plugin::AssertOS.....	78
SYNOPSIS.....	78
DESCRIPTION	78
ATTRIBUTES	78
METHODS	78
AUTHOR.....	79
KUDOS.....	79
LICENSE	79
SEE ALSO	79
Dist::Zilla::Plugin::PortabilityTests.....	80
VERSION.....	80
SYNOPSIS.....	80
DESCRIPTION	80
INSTALLATION	80
BUGS AND LIMITATIONS	80
AVAILABILITY	80
AUTHOR.....	80
COPYRIGHT AND LICENSE.....	80
Dist::Zilla::Plugin::Repository.....	81
VERSION.....	81
SYNOPSIS.....	81
DESCRIPTION	81
ATTRIBUTES	81
AUTHORS.....	81
COPYRIGHT AND LICENSE.....	81
Dist::Zilla::Plugin::Bugtracker	82
VERSION.....	82
SYNOPSIS.....	82
DESCRIPTION	82
FUNCTIONS.....	82
metadata.....	82
INSTALLATION	82
BUGS AND LIMITATIONS	82
AVAILABILITY	82

AUTHOR.....	83
COPYRIGHT AND LICENSE.....	83
Dist::Zilla::Plugin::HasVersionTests.....	84
VERSION.....	84
SYNOPSIS.....	84
DESCRIPTION.....	84
INSTALLATION.....	84
BUGS AND LIMITATIONS.....	84
AVAILABILITY.....	84
AUTHOR.....	84
COPYRIGHT AND LICENSE.....	84
Dist::Zilla::Plugin::Git::Commit.....	85
VERSION.....	85
SYNOPSIS.....	85
DESCRIPTION.....	85
METHODS.....	86
get_commit_message.....	86
AUTHOR.....	86
COPYRIGHT AND LICENSE.....	86
Dist::Zilla::Plugin::BumpVersion.....	87
VERSION.....	87
SYNOPSIS.....	87
AUTHOR.....	87
COPYRIGHT AND LICENSE.....	87
Dist::Zilla::Plugin::UploadToCPAN.....	88
VERSION.....	88
SYNOPSIS.....	88
DESCRIPTION.....	88
AUTHOR.....	88
COPYRIGHT AND LICENSE.....	88
Dist::Zilla::Plugin::CheckExtraTests.....	89
VERSION.....	89
SYNOPSIS.....	89
DESCRIPTION.....	89
SEE ALSO.....	89
AUTHOR.....	89
COPYRIGHT AND LICENSE.....	89
Dist::Zilla::Plugin::ShareDir.....	90
VERSION.....	90
SYNOPSIS.....	90
AUTHOR.....	90
COPYRIGHT AND LICENSE.....	90
Dist::Zilla::Plugin::Signature.....	91
VERSION.....	91
DESCRIPTION.....	91
ATTRIBUTES.....	91
AUTHOR.....	91
COPYRIGHT AND LICENSE.....	91
Dist::Zilla::Plugin::PodSyntaxTests.....	92
VERSION.....	92
DESCRIPTION.....	92
AUTHOR.....	92
COPYRIGHT AND LICENSE.....	92
Dist::Zilla::Plugin::NoTabsTests.....	93

DESCRIPTION	93
AUTHOR	93
COPYRIGHT AND LICENSE.....	93
Dist::Zilla::Plugin::MakeMaker	94
VERSION	94
DESCRIPTION	94
AUTHOR	94
COPYRIGHT AND LICENSE.....	94
Dist::Zilla::Plugin::Manifest	95
VERSION	95
DESCRIPTION	95
AUTHOR	95
COPYRIGHT AND LICENSE.....	95
Dist::Zilla::Plugin::Git::Check	96
VERSION	96
SYNOPSIS.....	96
DESCRIPTION	96
AUTHOR	96
COPYRIGHT AND LICENSE.....	96
Dist::Zilla::Plugin::Readme	97
VERSION	97
DESCRIPTION	97
AUTHOR	97
COPYRIGHT AND LICENSE.....	97
Dist::Zilla::Plugin::MetaYAML	98
VERSION	98
DESCRIPTION	98
ATTRIBUTES	98
filename.....	98
version.....	98
AUTHOR	98
COPYRIGHT AND LICENSE.....	98
Dist::Zilla::Plugin::Git::Push	99
VERSION	99
SYNOPSIS.....	99
DESCRIPTION	99
AUTHOR	99
COPYRIGHT AND LICENSE.....	99
Dist::Zilla::Plugin::EOLTests	100
DESCRIPTION	100
ATTRIBUTES	100
trailing_whitespace	100
AUTHOR	100
COPYRIGHT AND LICENSE.....	100
Dist::Zilla::Plugin::AutoPrereqs	101
VERSION	101
SYNOPSIS.....	101
DESCRIPTION	101
ATTRIBUTES	101
skip.....	101
CREDITS.....	101
AUTHOR	101
COPYRIGHT AND LICENSE.....	101
Dist::Zilla::Plugin::GatherDir	102

VERSION	102
DESCRIPTION	102
ATTRIBUTES	102
root	102
prefix	102
include_dotfiles	102
AUTHOR	103
COPYRIGHT AND LICENSE	103
Dist::Zilla::Plugin::ReadmeFromPod	104
SYNOPSIS	104
DESCRIPTION	104
AUTHORS	104
LICENSE AND COPYRIGHT	104
Dist::Zilla::Plugin::AutoVersion	105
VERSION	105
DESCRIPTION	105
ATTRIBUTES	105
major	105
format	105
AUTHOR	105
COPYRIGHT AND LICENSE	105
Dist::Zilla::Plugin::MetaNoIndex	106
VERSION	106
SYNOPSIS	106
DESCRIPTION	106
ATTRIBUTES	106
directory	106
file	106
package	106
namespace	106
METHODS	106
metadata	106
SEE ALSO	107
AUTHORS	107
COPYRIGHT AND LICENSE	107
Dist::Zilla::Plugin::CriticTests	108
VERSION	108
SYNOPSIS	108
DESCRIPTION	108
AUTHOR	108
COPYRIGHT AND LICENSE	108
Dist::Zilla::Plugin::CheckChangesHasContent	109
VERSION	109
SYNOPSIS	109
DESCRIPTION	109
SEE ALSO	109
AUTHOR	109
COPYRIGHT AND LICENSE	109
Dist::Zilla::Plugin::InlineFiles	111
VERSION	111
DESCRIPTION	111
AUTHOR	111
COPYRIGHT AND LICENSE	111
Dist::Zilla::Plugin::PodCoverageTests	112

VERSION	112
DESCRIPTION	112
AUTHOR	112
COPYRIGHT AND LICENSE.....	112
Dist::Zilla::Plugin::Git	113
VERSION	113
DESCRIPTION	113
SEE ALSO	113
AUTHOR	113
COPYRIGHT AND LICENSE.....	113
Dist::Zilla::Plugin::Prepender.....	115
VERSION	115
SYNOPSIS.....	115
DESCRIPTION	115
SEE ALSO	115
AUTHOR	116
COPYRIGHT AND LICENSE.....	116
Dist::Zilla::Plugin::ManifestSkip.....	117
VERSION	117
DESCRIPTION	117
ATTRIBUTES	117
skipfile.....	117
AUTHOR	117
COPYRIGHT AND LICENSE.....	117
Dist::Zilla::Plugin::AutoPrereq.....	118
VERSION	118
AUTHOR	118
COPYRIGHT AND LICENSE.....	118
Dist::Zilla::Plugin::GenerateFile.....	119
VERSION	119
SYNOPSIS.....	119
DESCRIPTION	119
Templating of the content	119
ATTRIBUTES	119
filename.....	119
content.....	119
is_template.....	119
AUTHOR	120
COPYRIGHT AND LICENSE.....	120
Dist::Zilla::Plugin::ExecDir.....	121
VERSION	121
SYNOPSIS.....	121
AUTHOR	121
COPYRIGHT AND LICENSE.....	121
Dist::Zilla::Plugin::FakeRelease	122
VERSION	122
SYNOPSIS.....	122
DESCRIPTION	122
AUTHOR	122
COPYRIGHT AND LICENSE.....	122
Dist::Zilla::Plugin::MetaJSON	123
VERSION	123
DESCRIPTION	123
ATTRIBUTES	123

filename.....	123
version.....	123
AUTHOR.....	123
COPYRIGHT AND LICENSE.....	123
Dist::Zilla::Plugin::KwaliteeTests.....	124
VERSION.....	124
SYNOPSIS.....	124
DESCRIPTION.....	124
INSTALLATION.....	124
BUGS AND LIMITATIONS.....	124
AVAILABILITY.....	124
AUTHOR.....	124
COPYRIGHT AND LICENSE.....	124
Dist::Zilla::Plugin::MetaProvides::FromFile.....	125
VERSION.....	125
ROLES.....	125
.....	125
PLUGIN FIELDS.....	125
file.....	125
type: required, ro, Str.....	125
reader_name.....	125
type: ClassName, ro.....	125
default: Config::INI::Reader.....	125
PRIVATE PLUGIN FIELDS.....	125
_reader.....	125
type: Object, ro, built from.....	125
ROLE SATISFYING METHODS.....	125
provides.....	125
signature: \$plugin->provides().....	125
returns: Array of.....	125
BUILDER METHODS.....	125
_build_reader.....	125
SEE ALSO.....	126
AUTHOR.....	126
COPYRIGHT AND LICENSE.....	126
Dist::Zilla::Plugin::MetaTests.....	127
VERSION.....	127
DESCRIPTION.....	127
AUTHOR.....	127
COPYRIGHT AND LICENSE.....	127
Appendix.....	127
COPYRIGHT AND LICENSE.....	127

Dist::Zilla - distribution builder; installer not included!

DESCRIPTION

Dist::Zilla builds distributions of code to be uploaded to the CPAN. In this respect, it is like ExtUtils::MakeMaker , Module::Build , or Module::Install . Unlike those tools, however, it is not also a system for installing code that has been downloaded from the CPAN. Since it's only run by authors, and is meant to be run on a repository checkout rather than on published, released code, it can do much more than those tools, and is free to make much more ludicrous demands in terms of prerequisites.

If you have access to the web, you can learn more and find an interactive tutorial at *dzil.org*. If not, try Dist::Zilla::Tutorial .

ATTRIBUTES

name

The name attribute (which is required) gives the name of the distribution to be built. This is usually the name of the distribution's main module, with the double colons (::) replaced with dashes. For example: Dist-Zilla.

version

This is the version of the distribution to be created.

abstract

This is a one-line summary of the distribution. If none is given, one will be looked for in the main_module of the dist.

main_module

This is the module where Dist::Zilla might look for various defaults, like the distribution abstract. By default, it's derived from the distribution name. If your distribution is Foo-Bar, and lib/Foo/Bar.pm exists, that's the main_module. Otherwise, it's the shortest-named module in the distribution. This may change!

You can override the default by specifying the file path explicitly, ie:

```
main_module = lib/Foo/Bar.pm
```

license

This is the Software::License object for this dist's license and copyright.

It will be created automatically, if possible, with the copyright_holder and copyright_year attributes. If necessary, it will try to guess the license from the POD of the dist's main module.

A better option is to set the license name in the dist's config to something understandable, like Perl_5.

authors

This is an arrayref of author strings, like this:

```
[  
  'Ricardo Signes <rjbs@cpan.org>',  
  'X. Ample, Jr <example@example.biz>',  
]
```

This is likely to change at some point in the near future.

files

This is an arrayref of objects implementing `Dist::Zilla::Role::File` that will, if left in this arrayref, be built into the dist.

Non-core code should avoid altering this arrayref, but sometimes there is not other way to change the list of files. In the future, the representation used for storing files *will be changed*.

root

This is the root directory of the dist, as a `Path::Class::Dir`. It will nearly always be the current working directory in which dzil was run.

is_trial

This attribute tells us whether or not the dist will be a trial release.

plugins

This is an arrayref of plugins that have been plugged into this `Dist::Zilla` object.

Non-core code *must not* alter this arrayref. Public access to this attribute *may go away* in the future.

distmeta

This is a hashref containing the metadata about this distribution that will be stored in `META.yml` or `META.json`. You should not alter the metadata in this hash; use a `MetaProvider` plugin instead.

prereqs

This is a `Dist::Zilla::Prereqs` object, which is a thin layer atop `CPAN::Meta::Prereqs`, and describes the distribution's prerequisites.

logger

This attribute stores a `Log::Dispatchouli::Proxy` object, used to log messages. By default, a proxy to the dist's Chrome is taken.

The following methods are delegated from the `Dist::Zilla` object to the logger:

- `log`

- log_debug
- log_fatal

METHODS

plugin_named

```
my $plugin = $zilla->plugin_named( $plugin_name );
```

plugins_with

```
my $roles = $zilla->plugins_with( -SomeRole );
```

This method returns an arrayref containing all the Dist::Zilla object's plugins that perform a the named role. If the given role name begins with a dash, the dash is replaced with "Dist::Zilla::Role::"

find_files

```
my $files = $zilla->find_files( $finder_name );
```

This method will look for a FileFinder-performing plugin with the given name and return the result of calling find_files on it. If no plugin can be found, an exception will be raised.

stash_named

```
my $stash = $zilla->stash_named( $name );
```

This method will return the stash with the given name, or undef if none exists. It looks for a local stash (for this dist) first, then falls back to a global stash (from the user's global configuration).

SUPPORT

There are usually people on irc.perl.org in #distzilla, even if they're idling.

The Dist::Zilla website has several valuable resources for learning to use Dist::Zilla.

There is a mailing list to discuss Dist::Zilla. You can join the list or browse the archives.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

Dist::Zilla::Tutorial - how to use this "Dist::Zilla" thing

SYNOPSIS

Dist::Zilla builds distributions to be uploaded to the CPAN. That means that the first thing you'll need is some code.

Once you've got that, you'll need to configure Dist::Zilla. Here's a simple dist.ini:

```
name      = Carbon-Dating
version  = 0.003
author   = Alan Smithee <asmithee@example.org>
license  = Perl_5
copyright_holder = Alan Smithee
```

```
[@Classic]
```

```
[Prereq]
```

```
App::Cmd          = 0.013
Number::Nary      = 0
Sub::Exporter     = 0.981
```

The topmost section configures Dist::Zilla itself. Here are some of the entries it expects:

```
name      - (required) the name of the dist being built
version   - (required) the version of the dist
abstract  - (required) a short description of the dist
author    - (optional) the dist author (you may have multiple entries for this)
license   - (required) the dist license; must be a Software::License::* name

copyright_holder - (required) the entity holding copyright on the dist
```

Some of the required values above may actually be provided by means other than the top-level section of the config. For example, VersionProvider plugins can set the version, and a line like this in the "main module" of the dist will set the abstract:

```
# ABSTRACT: a totally cool way to do totally great stuff
```

The main module is the module that shares the same name as the dist, in general.

Named sections load plugins, with the following rules:

If a section name begins with an equals sign, the rest of the section name is left intact and not expanded. If the section name begins with an at sign, it is prepended with 'Dist::Zilla::PluginBundle:.'. Otherwise, it is prepended with 'Dist::Zilla::Plugin:.'.

The values inside a section are given as configuration to the plugin. Consult each plugin's documentation for more information.

The "Classic" bundle, seen above, builds a fairly normal distribution. It rewrites tests from `./xt`, adds some information to POD, and builds a Makefile.PL. For more information, you can look at the docs for `Dist::Zilla::PluginBundle::Classic` and see the plugins it includes.

BUILDING YOUR DIST

Maybe we're getting ahead of ourselves, here. Configuring a bunch of plugins won't do you a lot of good unless you know how to use them to build your dist.

Dist::Zilla ships with a command called `dzil` that will get installed by default. While it can be extended to offer more commands, there are two really useful ones:

```
$ dzil build
```

The `build` command will build the distribution. Say you're using the configuration in the SYNOPSIS above. You'll end up with a file called `Carbon-Dating-0.004.tar.gz`. As long as you've done everything right, it will be suitable for uploading to the CPAN.

Of course, you should really test it out first. You can test the dist you'd be building by running another `dzil` command:

```
$ dzil test
```

This will build a new copy of your distribution and run its tests, so you'll know whether the dist that build would build is worth releasing!

HOW BUILDS GET BUILT

This is really more of a sketchy overview than a spec.

First, all the plugins that perform the `BeforeBuild` perform their `before_build` tasks.

The build root (where the dist is being built) is made.

The `FileGatherers` gather and inject files into the distribution, then the `FilePruners` remove some of them.

All the `FileMungers` get a chance to muck about with each file, possibly changing its name, content, or installability.

Now that the distribution is basically set up, it needs an install tool, like a Makefile.PL. All the `InstallTool`-performing plugins are used to do whatever is needed to make the dist installable.

Everything is just about done. The files are all written out to disk and the `AfterBuild` plugins do their thing.

THE GLORIOUS FUTURE

In the glorious future of Dist::Zilla, another phase in the process will exist: `release`. By running `dzil release`, you'll be able to test your distribution, build a tarball of it, and upload it to the CPAN. Plugins will be able to do things like check your version control system to make sure you're releasing a new version and that you tag the version you've just uploaded. It will update your Changelog file, too, making sure that you don't need to know what your next version number will be before releasing.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

Dist::Zilla::Plugin::Prereqs - list simple prerequisites

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[Prereqs]
Foo::Bar = 1.002
MRO::Compat = 10
Sub::Exporter = 0
```

Which is equivalent to specifying prerequisites for the Runtime phase:

```
[Prereqs / RuntimeRequires]
Foo::Bar = 1.002
MRO::Compat = 10
Sub::Exporter = 0
```

See Phases for the full list of supported phases.

DESCRIPTION

This module adds "fixed" prerequisites to your distribution. These are prereqs with a known, fixed minimum version that doesn't change based on platform or other conditions.

You can specify prerequisites for different phases and kinds of relationships. In RuntimeRequires, the phase is Runtime and the relationship is Requires. These are described in more detail in the CPAN::Meta specification.

The phases are:

- configure
- build
- test
- runtime
- develop

The relationship types are:

- requires
- recommends

- suggests
- conflicts

Not all of these phases are useful for all tools, especially tools that only understand version 1.x CPAN::Meta files.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::RemovePrereqs **- a plugin to remove gathered prereqs**

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[RemovePrereqs]  
remove = Foo::Bar  
remove = MRO::Compat
```

This will remove any prerequisite of any type from any prereq phase. This is useful for eliminating incorrectly detected prereqs.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MinimumVersionTests **- release tests for minimum required versions**

VERSION

version 1.100700

SYNOPSIS

In dist.ini:

```
[MinimumVersionTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files

```
xt/release/minimum-version.t - a standard Test::MinimumVersion test
```

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::ReportVersions - Write a test that reports used module versions

VERSION

version 1.101700

SYNOPSIS

In dist.ini:

```
[ReportVersions]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following file:

```
t/000-report-versions.t
```

The 000 prefix is chosen so it runs first to make sure it shows up in CPAN tester reports.

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-ReportVersions/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-ReportVersions/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::GatherDir::Template **- gather all the files in a directory and use them as templates**

VERSION

version 4.102345

DESCRIPTION

This is a very, very simple FileGatherer plugin. It looks in the directory named in the root attribute and adds all the files it finds there. If the root begins with a tilde, the tilde is replaced with the current user's home directory according to File::HomeDir .

It is meant to be used when minting dists with dzil new, but could be used in building existing dists, too.

AUTHOR

Ricardo SIGNES <tjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::GithubMeta - Automatically include GitHub meta information in META.yml

VERSION

version 0.10

SYNOPSIS

```
# in dist.ini
```

```
[GithubMeta]
```

```
# to override the homepage
```

```
[GithubMeta]
```

```
homepage = http://some.sort.of.url/project/
```

```
# to override the github remote repo (defaults to 'origin')
```

```
[GithubMeta]
```

```
remote=github
```

DESCRIPTION

Dist::Zilla::Plugin::GithubMeta is a Dist::Zilla plugin to include GitHub <http://github.com> meta information in META.yml.

It automatically detects if the distribution directory is under git version control and whether the origin is a GitHub repository and will set the repository and homepage meta in META.yml to the appropriate URLs for GitHub.

Based on Module::Install::GithubMeta which was based on Module::Install::Repository by Tatsuhiko Miyagawa

ATTRIBUTES

remote

The GitHub remote repo can be overridden with this attribute. If not provided, it defaults to origin.

homepage

You may override the homepage setting by specifying this attribute. This should be a valid URL as understood by MooseX::Types::URI .

METHODS

metadata

Required by Dist::Zilla::Role::MetaProvider

SEE ALSO

Dist::Zilla

AUTHOR

Chris Williams <chris@bingosnet.co.uk>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Chris Williams and Tatsuhiko Miyagawa.

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

Dist::Zilla::Plugin::InstallGuide - Build an INSTALL file

VERSION

version 1.101461

SYNOPSIS

In dist.ini:

```
[InstallGuide]
```

DESCRIPTION

This plugin adds a very simple `INSTALL` file to the distribution, telling the user how to install this distribution.

You should use this plugin in your `Dist::Zilla` configuration after `[MakeMaker]` or `[ModuleBuild]` so that it can determine what kind of distribution you are building and which installation instructions are appropriate.

FUNCTIONS

setup_installer

Builds and writes the `INSTALL` file.

INSTALLATION

See `perlmodinstall` for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-InstallGuide>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-InstallGuide/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-InstallGuide/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::License - output a LICENSE file

VERSION

version 4.102345

DESCRIPTION

This plugin adds a LICENSE file containing the full text of the distribution's license, as produced by the fulltext method of the dist's Software::License object.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::PodVersion - add a VERSION head1 to each Perl document

VERSION

version 4.102345

DESCRIPTION

This plugin adds a =head1 VERSION section to most perl files in the distribution, indicating the version of the dist being built. This section is added after =head1 NAME. If there is no such section, the version section will not be added.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::TemplateModule - a simple module-from-template plugin

VERSION

version 4.102345

DESCRIPTION

This is a ModuleMaker used for creating new Perl modules files when minting a new dist with dzil new. It uses Text::Template (via Dist::Zilla::Role::TextTemplate) to render a template into a Perl module. The template is given two variables for use in rendering: \$name, the module name; and \$dist, the Dist::Zilla object. The module is always created as a file under ./lib.

By default, the template looks something like this:

```
use strict;
use warnings;
package {{ $name }};

1;
```

ATTRIBUTES

template

The template parameter may be given to the plugin to provide a different filename, absolute or relative to the build root.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Prereq - (DEPRECATED) the old name of the Prereqs plugin

VERSION

version 4.102345

SYNOPSIS

This plugin extends [Prereqs] and adds nothing. It is the old name for Prereqs, and will be removed in a few versions.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::FinderCode - a callback-based FileFinder plugin

VERSION

version 4.102345

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git::Init - initialize git repository on dzil new

VERSION

version 1.102020

DESCRIPTION

This plugin initializes a git repository when a new distribution is created with dzil new.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::PkgVersion - add a \$VERSION to your packages

VERSION

version 4.102345

SYNOPSIS

in dist.ini

```
[PkgVersion]
```

DESCRIPTION

This plugin will add lines like the following to each package in each Perl module or program (more or less) within the distribution:

```
BEGIN {  
    $MyModule::VERSION = 0.001;  
}
```

...where 0.001 is the version of the dist, and MyModule is the name of the package being given a version. (In other words, it always uses fully-qualified names to assign versions.)

It will skip any package declaration that includes a newline between the package keyword and the package name, like:

```
package  
    Foo::Bar;
```

This sort of declaration is also ignored by the CPAN toolchain, and is typically used when doing monkey patching or other tricky things.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git::Tag - tag the new version

VERSION

version 1.102020

SYNOPSIS

In your dist.ini:

```
[Git::Tag]
tag_format = v%v          ; this is the default
tag_message = v%v        ; this is the default
```

DESCRIPTION

Once the release is done, this plugin will record this fact in git by creating a tag. By default, it makes an annotated tag. You can set the `tag_message` attribute to change the message. If you set `tag_message` to the empty string, it makes a lightweight tag.

It also checks before the release to ensure the tag to be created doesn't already exist. (You would have to manually delete the existing tag before you could release the same version again, but that is almost never a good idea.)

Plugin options

The plugin accepts the following options:

- * `tag_format` - format of the tag to apply. Defaults to `v%v`, see Formatting options below.
- * `tag_message` - format of the commit message. Defaults to `v%v`, see Formatting options below. Use `tag_message =` to create a lightweight tag.
- * `branch` - which branch to tag. Defaults to current branch.

Formatting options

Some plugin options allow you to customize the tag content. You can use the following codes at your convenience:

`%{dd-MMM-yyyy}d`

The current date. You can use any CLDR format supported by `DateTime`. A bare `%d` means `%{dd-MMM-yyyy}d`.

`%n`

A newline

`%N`

The distribution name

%v

The distribution version

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::BumpVersionFromGit

- provide a version number by bumping the last git release tag

VERSION

version 0.007

SYNOPSIS

In your F<dist.ini>:

```
[BumpVersionFromGit]
first_version = 0.001      ; this is the default
version_regexp = ^v(.+)$  ; this is the default
```

DESCRIPTION

This does the Dist::Zilla::Role::VersionProvider role. It finds the last version number from tags and increments it as the new version used by Dist::Zilla.

The plugin accepts the following options:

- `first_version` - if the repository has no tags at all, this version is used as the first version for the distribution. It defaults to "0.001".
- `version_regexp` - regular expression that matches a tag containing a version. It should capture the version into \$1. Defaults to `^v(.+)$` which matches the default tag from Dist::Zilla::Plugin::Git::Tag

You can also set the `V` environment variable to override the new version. This is useful if you need to bump to a specific version. For example, if the last tag is 0.005 and you want to jump to 1.000 you can set `V = 1.000`.

NOTE-- this module is a stop gap while Dist::Zilla is enhanced to allow more sophisticated version number manipulation and may be deprecated in the future once those changes are complete.

SEE ALSO

- Dist::Zilla::Plugin::VersionFromPrev and Dist::Zilla::Plugin::Git::LastVersion do something similar but in what I find to be a more complicated way of doing things

AUTHOR

David Golden <dagolden@cpan.org>

COPYRIGHT AND LICENSE

This software is Copyright (c) 2010 by David Golden.

This is free software, licensed under:

The Apache License, Version 2.0, January 2004

Dist::Zilla::Plugin::TaskWeaver - a PodWeaver plugin used to build Task distributions

VERSION

version 0.100890

DESCRIPTION

The TaskWeaver plugin acts just like the PodWeaver plugin, but gets its claws just a bit into your Pod::Weaver configuration and then uses them to figure out prerequisites and grouping for building a Task distribution.

The Task:: namespace is used for libraries that do not have any code of their own, but are just ways of getting a lot of other libraries installed at once. In other words, they're just prerequisites with no actual logic.

TaskWeaver expects that your .pm file will have Pod like the following:

```
=pkggroup Modules That are Useful

=pkg Sub::Exporter 0.901 first version with teeth

=pkg Sub::Import

=pkggroup Modules That are Useless

=pkg Acme::ProgressBar 1.10
```

The =pkggroup directives start groups of prerequisites. You must have at least one =pkggroup (although that may change). The =pkg directives list specific directives, and are in the following format:

```
=pkg Package::Name min_version reason
```

Both min_version and reason are optional, although you can't give a reason without giving a reason. If a reason is given, it will be included in the Pod to explain why the specific version is required.

WARNING

TaskWeaver works, but relies on doing some pretty evil stuff. It may substantially change its method of operation in the future, but its expectations from your Pod should not change.

AUTHOR

Ricardo Signes <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::ModuleInstall - Build Module::Install based Distributions with Dist::Zilla

VERSION

version 0.01002219

SYNOPSIS

dist.ini

```
[ModuleInstall]
[MakeMaker::Tests] # use the testing part of eumm.
```

DESCRIPTION

This module will create a Makefile.PL for installing the dist using Module::Install .

It is at present a very minimal feature set, but it works.

AUTHOR

Kent Fredric <kentnl@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Kent Fredric.

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

Dist::Zilla::Plugin::ChangelogFromGit - build **CHANGES** from git commits and tags

VERSION

version 0.002

SYNOPSIS

In your dist.ini:

```
[ChangelogFromGit]
max_age      = 365
tag_regexp   = ^v\d+_\d+$
file_name    = CHANGES
wrap_column  = 74
```

The example values are the defaults.

DESCRIPTION

This Dist::Zilla plugin writes a CHANGES file that contains formatted commit information from recent git logs.

This plugin has the following configuration variables:

* max_age

It may be impractical to include the full change log in a mature project's distribution. "max_age" limits the changes to the most recent ones within a number of days. The default is about one year.

Include two years of changes:

```
max_age = 730
```

* tag_regexp

This plugin breaks the changelog into sections delineated by releases, which are defined by release tags. "tag_regexp" may be used to focus only on those tags that follow a particular release tagging format. Some of the author's repositories contain multiple projects, each with their own specific release tag formats, so that changelogs can focus on particular projects' tags. For instance, POE::Test::Loops' release tags may be specified as:

```
tag_regexp = ^ptl-
```

* file_name

Everyone has a preference for their change logs. If you prefer lowercase in your change log file names, you might specify:

```
file_name = Changes
```

* wrap_column

Changes from different contributors tend to vary in format. This plugin uses Text::Wrap to normalize the width of commit messages. The "wrap_column" parameter may be used to alter the reformatted line width. If 74 is too short, one might specify:

```
wrap_column = 78
```

Subversion and CVS

This plugin is almost entirely a copy-and-paste port of a command-line tool I wrote a while ago. I also have tools to generate change logs from CVS and Subversion commits. If anyone is interested, plugins for these other version control systems should be about an hour's work apiece.

AUTHOR

Rocco Caputo <rcaputo@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Rocco Caputo.

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

Dist::Zilla::Plugin::PodWeaver - weave your Pod together from configuration and Dist::Zilla VERSION

version 3.101640

DESCRIPTION

[PodWeaver] is the bridge between Dist::Zilla and Pod::Weaver . It rips apart your kinda-Pod and reconstructs it as boring old real Pod.

METHODS

weaver

This method returns the Pod::Weaver object to be used. The current implementation builds a new weaver on each invocation, because one or two core Pod::Weaver plugins cannot be trusted to handle multiple documents per plugin instance. In the future, when that is fixed, this may become an accessor of an attribute with a builder. Until this is clearer, use caution when modifying this method in subclasses.

CONFIGURATION

If the `config_plugin` attribute is given, it will be treated like a Pod::Weaver section heading. For example, `@Default` could be given.

Otherwise, if a file matching `./weaver.*` exists, Pod::Weaver will be told to look for configuration in the current directory.

Otherwise, it will use the default configuration.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git::CommitBuild - checkin build results on separate branch

VERSION

version 1.102020

SYNOPSIS

In your dist.ini:

```
[Git::CommitBuild]
    ; these are the defaults
branch = build/%b
message = Build results of %h (on %b)
```

DESCRIPTION

Once the build is done, this plugin will commit the results of the build to a branch that is completely separate from your regular code branches (i.e. with a different root commit). This potentially makes your repository more useful to those who may not have Dist::Zilla and all of its dependencies installed.

The plugin accepts the following options:

- A single formatting code (%b) is defined for this attribute and will be substituted with the name of the current branch in your git repository.
- Same as branch, but commit the build content only after a release. No default, meaning no release branch.
- This option supports three formatting codes:
 - * %b - Name of the current branch
 - * %H - Commit hash
 - * %h - Abbreviated commit hash

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::PerlTidy - PerlTidy in Dist::Zilla

VERSION

version 0.09

METHODS

`munge_file`

Implements the required `munge_file` method for the `Dist::Zilla::Role::FileMunger` role, munging each Perl file it finds. Files whose names do not end in `.pm`, `.pl`, or `.t`, or whose contents do not begin with `#!/perl` are left alone.

SYNOPSIS

```
# dist.ini
[PerlTidy]

# or
[PerlTidy]
perltidyrc = xt/.perltidyrc
```

DEFAULTS

If you do not specify a specific `perltidyrc` in `dist.ini` it will try to use the same defaults as `Perl::Tidy`.

SEE ALSO

`Perl::Tidy`

AUTHORS

Fayland Lam <fayland@gmail.com>
Mark Gardner <mjgardner@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Fayland Lam.

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

Dist::Zilla::Plugin::MatchManifest - Ensure that MANIFEST is correct

VERSION

This document describes version 0.03 of Dist::Zilla::Plugin::MatchManifest, released April 14, 2010.

SYNOPSIS

[MatchManifest]

DESCRIPTION

If included, this plugin will ensure that the distribution contains a MANIFEST file and that its contents match the files collected by Dist::Zilla. If not, it will display the differences and (if STDIN & STDOUT are TTYs) offer to update the MANIFEST.

As I see it, there are 2 problems that a MANIFEST can protect against:

A file I don't want to distribute winds up in the tarball

A file I did want to distribute gets left out of the tarball

By keeping your MANIFEST under source control and using this plugin to make sure it's kept up to date, you can protect yourself against both problems.

CONFIGURATION AND ENVIRONMENT

Dist::Zilla::Plugin::MatchManifest requires no configuration files or environment variables.

INCOMPATIBILITIES

None reported.

BUGS AND LIMITATIONS

No bugs have been reported.

AUTHOR

Christopher J. Madsen <perl AT cjmweb.net>

Please report any bugs or feature requests to <bug-Dist-Zilla-Plugin-MatchManifest AT rt.cpan.org>, or through the web interface at <http://rt.cpan.org/Public/Bug/Report.html?Queue=Dist-Zilla-Plugin-MatchManifest>

You can follow or contribute to Dist-Zilla-Plugin-MatchManifest's development at <http://github.com/madsen/dist-zilla-plugin-matchmanifest>.

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Christopher J. Madsen.

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

DISCLAIMER OF WARRANTY

BECAUSE THIS SOFTWARE IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE SOFTWARE, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE SOFTWARE AS PERMITTED BY THE ABOVE LICENSE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE SOFTWARE TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Dist::Zilla::Plugin::PkgDist - add a \$DIST to your packages

VERSION

version 4.102345

DESCRIPTION

This plugin will add a line like the following to each package in each Perl module or program (more or less) within the distribution:

```
{ our $DIST = 'My-CPAN-Dist'; } # where 'My-CPAN-Dist' is your dist name
```

It will skip any package declaration that includes a newline between the package keyword and the package name, like:

```
package  
    Foo::Bar;
```

This sort of declaration is also ignored by the CPAN toolchain, and is typically used when doing monkey patching or other tricky things.

AUTHOR

Ricardo SIGNES <tjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::TestRelease - extract archive and run tests before releasing the dist

VERSION

version 4.102345

DESCRIPTION

This plugin runs before a release happens. It will extract the to-be-released archive into a temporary directory and use the TestRunner plugins to run its tests. If the tests fail, the release is aborted and the temporary directory is left in place. If the tests pass, the temporary directory is cleaned up and the release process continues.

CREDITS

This plugin was originally contributed by Christopher J. Madsen.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaProvides - Generating and Populating 'provides' in your META.yml

VERSION

version 1.10027802

SYNOPSIS

In your projects dist.ini

```
[MetaProvides::Class]
inherit_version = 0      ;optional flag
inherit_missing = 0     ;optional flag

[MetaProvides::Package]
inherit_version = 0      ;optional flag
inherit_missing = 0     ;optional flag

[MetaProvides::FromFile]
inherit_version = 0      ;optional flag
inherit_missing = 0     ;optional flag
file = some_file.ini    ;mandatory flag
reader_name = Config::INI::Reader ;optional flag
```

And then in some_file.ini

```
[Imaginary::Package]
file = lib/Imaginary/Package.pm ;mandatory flag
version = 3.1415                ;optional flag, subject to rules in dist.ini
```

DESCRIPTION

This Distribution Contains a small bundle of plugins for various ways of populating the META.yml that is built with your distribution.

The initial reason for this is due to stuff that uses MooseX::Declare style class definitions not being parseable by many tools upstream, so this is here to cover this problem by defining it in the metadata.

COMPONENT SUMMARY

::Class

Scans Dist::Zilla's .pm files and tries to identify classes using Class::Discover .

Dist::Zilla::Plugin::MetaProvides::Class

::Package

Scans Dist::Zilla's .pm files and tries to identify more traditional packages using a combination of Module::Extract::VERSION and Module::Extact::Namespaces .

Dist::Zilla::Plugin::MetaProvides::Package

::FromFile

In the event both of the above don't work for your needs, pull in hand-crafted metadata from a specified file.

Dist::Zilla::Plugin::MetaProvides::FromFile

OPTION SUMMARY

inherit_version

At the time this plugin runs to collect metadata from files, the mungers won't have run yet to inject custom versions into files in the various locations.

If you want the versions reported in the provides list to be consistent with the ones actually in the files, you will need to use this option in its enabled state.

IE: Generally, if you are using version munging, you *WILL* want this flag set to 1.

values

* '0'

Do not inherit version from Dist::Zilla

* '1' *[default]*

Inherit version from Dist::Zilla

inherit_version in Dist::Zilla::Role::MetaProvider::Provider

inherit_missing

If for whatever reason you want to actually use the versions found in the modules where present, and fall back to the value from Dist::Zilla .

inherit_version will need to be turned off (0) for this to be effective.

values

* '0'

Do not inherit version from Dist::Zilla when one is missing.

* '1' *[default]*

Inherit version from Dist::Zilla when one is missing.

inherit_missing in Dist::Zilla::Role::MetaProvider::Provider

file

(Dist::Zilla::Plugin::MetaProvides::FromFile)

This is a mandatory parameter that points to the file that contains manually (or otherwise) crafted metadata to be integrated into your final META.yml

File Must exist.

reader_name

(Dist::Zilla::Plugin::MetaProvides::FromFile)

This parameter is by default Config::INI::Reader , but it can be infact anything that meets the following criteria.

- * Can be initialized an instance of
- * has a read_file method on the instance
- * read_file can take the parameter 'file'
- * read_file can return a hashref matching the following structure


```
{ 'Package::Name' => {
    'file' => '/path/to/file',
    'version' => 0.1,
  }}
```

AUTHOR

Kent Fredric <kentnl@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Kent Fredric.

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

Dist::Zilla::Plugin::CheckChangeLog - Dist::Zilla with Changes check VERSION

version 0.01

SYNOPSIS

```
# dist.ini  
[CheckChangeLog]
```

```
# or  
[CheckChangeLog]  
filename = Changes.pod
```

DESCRIPTION

The code is mostly a copy-paste of ShipIt::Step::CheckChangeLog

AUTHOR

Fayland Lam <fayland@gmail.com>

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Fayland Lam.

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

Dist::Zilla::Plugin::CompileTests - common tests to check syntax of your modules

VERSION

version 1.100740

SYNOPSIS

In your dist.ini:

```
[CompileTests]
skip          = Test$
fake_home    = 1
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

- * t/00-compile.t - a standard test to check syntax of bundled modules

This test will find all modules and scripts in your dist, and try to compile them one by one. This means it's a bit slower than loading them all at once, but it will catch more errors.

This plugin accepts the following options:

- * skip: a regex to skip compile test for modules matching it. The match is done against the module name (Foo::Bar), not the file path (lib/Foo/Bar.pm).

- * fake_home: a boolean to indicate whether to fake \$ENV{HOME}. This may be needed if your module unilaterally creates stuff in homedir: indeed, some cpan testers will smoke test your dist with a read-only home directory. Default to false.

SEE ALSO

You can also look for information on this module at:

- * AnnoCPAN: Annotated CPAN documentation

<http://annocpan.org/dist/Dist-Zilla-Plugin-CompileTests>

- * CPAN Ratings

<http://cpanratings.perl.org/d/Dist-Zilla-Plugin-CompileTests>

- * Open bugs

<http://rt.cpan.org/NoAuth/Bugs.html?Dist=Dist-Zilla-Plugin-CompileTests>

* Git repository

<http://github.com/jquelin/dist-zilla-plugin-compiletests.git>.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::ExtraTests - rewrite ./xt tests to ./t tests with skips

VERSION

version 4.102345

DESCRIPTION

This plugin rewrites tests found in the following directories:

```
./xt/author - tests for author testing (env AUTHOR_TESTING is true)
./xt/release - tests for pre-release testers (env RELEASE_TESTING is true)
./xt/smoke - tests for automated testers (env AUTOMATED_TESTING is true)
```

The tests are renamed and moved to ./t, and they are rewritten to include some simple Perl code to skip all included tests if the correct env vars are not set.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

the timezone to use when generating the date; defaults to *local*

The module allows the following printf-like format codes in the format:

- v - the version of the dist
- d - the CLDR format for DateTime
- n - a newline
- t - a tab

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::UnusedVarsTests - Release tests for unused variables

VERSION

version 1.100860

SYNOPSIS

In dist.ini:

```
[UnusedVarsTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following file:

```
xt/release/unused-vars.t - a standard Test::Vars test
```

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-UnusedVarsTests>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-UnusedVarsTests/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-UnusedVarsTests/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::Homepage - Automatically sets the homepage URL

VERSION

version 1.101420

SYNOPSIS

In dist.ini:

```
[Homepage]
```

DESCRIPTION

This plugin sets the distribution's homepage URL as metadata.

FUNCTIONS

metadata

Sets the homepage URL in the distribution's metadata.

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-Homepage>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-Homepage/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-Homepage/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::PruneFiles - prune arbitrary files from the dist

VERSION

version 4.102345

SYNOPSIS

This plugin allows you to explicitly prune some files from your distribution. You can either specify the exact set of files (with the "filenames" parameter) or provide the regular expressions to check (using "match").

This is useful if another plugin (maybe a FileGatherer) adds a bunch of files, and you only want a subset of them.

In your dist.ini:

```
[PruneFiles]
filenames = xt/release/pod-coverage.t ; pod coverage tests are for jerks
```

```
match      = ^test_data/*
match      = ^test.cvs$
```

ATTRIBUTES

filenames

This is an arrayref of filenames to be pruned from the distribution.

matches

This is an arrayref of regular expressions and files matching any of them, will be pruned from the distribution.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::PruneCruft - prune stuff that you probably don't mean to include

VERSION

version 4.102345

SYNOPSIS

This plugin tries to compensate for the stupid crap that turns up in your working copy, removing it before it gets into your dist and screws everything up.

In your dist.ini:

```
[PruneCruft]
```

If you would like to exclude certain exclusions, use the except option (it can be specified multiple times):

```
[PruneCruft]
except = \.gitignore
except = t/*/\.keep$
```

AUTHOR

Ricardo SIGNES <tjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaProvides::Pack - Extract namespaces/version from traditional packages for provides

VERSION

version 1.10027802

ROLES

Dist::Zilla::Role::MetaProvider::Provider

ROLE SATISFYING METHODS

provides

A conformant function to the `Dist::Zilla::Role::MetaProvider::Provider` Role.

signature: \$plugin->provides()

**returns: Array of
Dist::Zilla::MetaProvides::ProvideRecord**

PRIVATE METHODS

_packages_for
**signature: \$plugin->_packages_for(\$filename,
\$file_content)**

**returns: Array of
Dist::Zilla::MetaProvides::ProvideRecord**

. SEE ALSO

AUTHOR

Kent Fredric <kentnl@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Kent Fredric.

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

Dist::Zilla::Plugin::ModuleBuild **- build a Build.PL that uses** **Module::Build**

VERSION

version 4.102345

DESCRIPTION

This plugin will create a Build.PL for installing the dist using Module::Build .

ATTRIBUTES

mb_version

Optional: Specify the minimum version of Module::Build to depend on.

Defaults to 0.3601

mb_class

Optional: Specify the class to use to create the build object. Defaults to Module::Build itself. If another class is specified, use lib 'inc' is also added to the Build.PL file.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::ModuleShareDirs **- install a directory's contents** **as module-based "ShareDir"** **content**

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[ModuleShareDirs]
Foo::Bar = shares/foo_bar
Foo::Baz = shares/foo_baz
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::ConfirmRelease - prompt for confirmation before releasing

VERSION

version 4.102345

DESCRIPTION

This plugin prompts the author whether or not to continue before releasing the distribution to CPAN. It gives authors a chance to abort before they upload.

The default is "no", but you can set the environment variable `DZIL_CONFIRMRELEASE_DEFAULT` to "yes" if you just want to hit enter to release.

This plugin uses `ExtUtils::MakeMaker::prompt()`, so setting `PERL_MM_USE_DEFAULT` to a true value will accept the default without prompting.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaProvides::Class

- Scans Dist::Zilla's .pm files and tries to identify classes using Class::Discover.

VERSION

version 1.10027802

ROLES

Dist::Zilla::Role::MetaProvider::Provider

ROLE SATISFYING METHODS

provides

A conformant function to the Dist::Zilla::Role::MetaProvider::Provider Role.

signature: \$plugin->provides()

**returns: Array of
Dist::Zilla::MetaProvides::ProvideRecord**

PRIVATE METHODS

_classes_for

**signature: \$plugin->_classes_for(\$filename,
\$file_content)**

**returns: Array of
Dist::Zilla::MetaProvides::ProvideRecord**

. SEE ALSO

AUTHOR

Kent Fredric <kentnl@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Kent Fredric.

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

Dist::Zilla::Plugin::MetaConfig - summarize Dist::Zilla configuration into distmeta VERSION

version 4.102345

DESCRIPTION

This plugin adds a top-level `x_Dist_Zilla` key to the distmeta for the distribution. It describe the `Dist::Zilla` version used as well as all the plugins used. Each plugin's name, package, and version will be included. Plugins may augment their implementation of the `Dist::Zilla::Role::ConfigDumper` role methods to add more data to this dump.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Twitter - Twitter when you release with Dist::Zilla

VERSION

version 0.007

SYNOPSIS

In your dist.ini:

```
[Twitter]
hash_tags = #foo
```

In your .netrc:

```
machine api.twitter.com
  login YOUR_TWITTER_USER_NAME
  password YOUR_TWITTER_PASSWORD
```

DESCRIPTION

This plugin will use Net::Twitter with the login and password in your .netrc file to send a release notice to Twitter. By default, it will include a link to your README file as extracted on a fast CPAN mirror. This works very nicely with Dist::Zilla::Plugin::ReadmeFromPod .

The default configuration is as follows:

```
[Twitter]
tweet_url = http://cpan.cpan testers.org/authors/id/{{ $AUTHOR_PATH }}/{{ $DIST }}-{{ $VERSION }}
tweet = Released {{ $DIST }}-{{ $VERSION }} {{ $URL }}
```

The tweet_url is shortened with WWW::Shorten::TinyURL and appended to the tweet message. The following variables are available for substitution in the URL and message templates:

```
DIST          # Foo-Bar
VERSION       # 1.23
TARBALL       # Foo-Bar-1.23.tar.gz
AUTHOR_UC     # JOHNDOE
AUTHOR_LC     # johndoe
AUTHOR_PATH   # J/JO/JOHNDOE
URL           # TinyURL
```

You must be using the UploadToCPAN plugin for this plugin to determine your CPAN author ID.

You can use the hash_tags option to append hash tags (or anything, really) to the end of the message generated from tweet.

[Twitter]
hash-tags = #perl #cpan #foo

AUTHOR

David Golden <dagolden@cpan.org>

COPYRIGHT AND LICENSE

This software is Copyright (c) 2010 by David Golden.

This is free software, licensed under:

The Apache License, Version 2.0, January 2004

Dist::Zilla::Plugin::MetaResources **- provide arbitrary "resources"** **for distribution metadata**

VERSION

version 4.102345

DESCRIPTION

This plugin adds resources entries to the distribution's metadata.

```
[MetaResources]
homepage          = http://example.com/~dude/project.asp
bugtracker.web    = http://rt.cpan.org/NoAuth/Bugs.html?Dist=Project
bugtracker.mailto = bug-project@rt.cpan.org
repository.url    = git://github.com/dude/project.git
repository.web    = http://github.com/dude/project
repository.type   = git
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::DistINI - a plugin to add a dist.ini to newly-minted dists

VERSION

version 4.102345

DESCRIPTION

This plugins produces a dist.ini file in a new dist, specifying the required core attributes from the dist being minted.

This plugin is dead simple and pretty stupid, but should get better as dist minting facilities improve. For example, it will not specify any plugins.

In the meantime, you may be happier with a dist.ini template.

ATTRIBUTES

append_file

This parameter may be a filename in the profile's directory to append to the generated dist.ini with things like plugins. In other words, if you make this file, called plugins.ini:

```
[@Basic]
[NextRelease]
[@Git]
```

...and your profile.ini includes:

```
[DistINI]
append_file = plugins.ini
```

...then the generated dist.ini in a newly-minted dist will look something like this:

```
name      = My-New-Dist
author    = E. Xavier Ample <example@example.com>
license   = Perl_5
copyright_holder = E. Xavier Ample
copyright_year   = 2010
```

```
[@Basic]
[NextRelease]
[@Git]
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::AssertOS - Require that our distribution is running on a particular OS

SYNOPSIS

```
# In dist.ini - It is important that AssertOS follows MakeMaker
```

```
[MakeMaker]
```

```
[AssertOS]
```

```
os = Linux
```

```
os = FreeBSD
```

```
os = cygwin
```

The resultant distribution will die at Makefile.PL unless the platform the code is running on is Linux, FreeBSD or Cygwin.

DESCRIPTION

Dist::Zilla::Plugin::AssertOS is a Dist::Zilla plugin that integrates Devel::AssertOS so that CPAN authors may easily stipulate which particular OS environments their distributions may be built and installed on.

The author specifies which OS or OS families are supported. The necessary Devel::AssertOS files are copied to the inc/ directory and Makefile.PL is mungled to include the necessary incantation.

On the module user side, the bundled inc/ Devel::AssertOS determines whether the current environment is supported or not and will die accordingly.

As this plugin mungles the Makefile.PL it is imperative that it is specified in dist.ini AFTER [MakeMaker].

This plugin also automatically adds the no_index metadata so that inc/ is excluded from PAUSE indexing. If you use Dist::Zilla::Plugin::MetaNoIndex , there may be conflicts.

ATTRIBUTES

os

Specify as many times as wanted the OS that you wish your distribution to work with. See Devel::AssertOS and Devel::CheckOS for what may be given.

METHODS

These are required by the roles that this plugin uses.

mvp_multivalue_args

gather_files

Required by Dist::Zilla::Role::FileGatherer .

setup_installer

Required by Dist::Zilla::Role::InstallTool .

metadata

Required by Dist::Zilla::Role::MetaProvider .

AUTHOR

Chris BinGOs Williams

Based on use-devel-assertos by David Cantrell

KUDOS

Thanks to Ricardo Signes, not only for Dist::Zilla , but for explaining Dist::Zilla::Role::InstallTool 's place in the build process. This made this plugin possible.

LICENSE

Copyright © Chris Williams and David Cantrell

This module may be used, modified, and distributed under the same terms as Perl itself. Please see the license that came with your Perl distribution for details.

SEE ALSO

Dist::Zilla

Devel::AssertOS

Devel::CheckOS

Dist::Zilla::Plugin::PortabilityTests - Release tests for portability

VERSION

version 1.101420

SYNOPSIS

In dist.ini:

```
[PortabilityTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following file:

```
xt/release/portability.t - a standard Test::Portability::Files test
```

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-PortabilityTests>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-PortabilityTests/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-PortabilityTests/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::Repository - Automatically sets repository URL from svn/svk/Git checkout for Dist::Zilla

VERSION

version 0.13

SYNOPSIS

```
# dist.ini  
[Repository]
```

DESCRIPTION

The code is mostly a copy-paste of Module::Install::Repository

ATTRIBUTES

* git_remote

This is the name of the remote to use for the public repository (if you use Git). By default, unsurprisingly, to origin.

* github_http

If the remote is a GitHub repository, uses the http url (<http://github.com/fayland/dist-zilla-plugin-repository>) rather than the actual clonable url (<git://github.com/fayland/dist-zilla-plugin-repository.git>). Defaults to true.

AUTHORS

Fayland Lam <fayland@gmail.com>
Ricardo SIGNES <rjbs@cpan.org>
Moritz Onken <onken@netcubed.de>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Fayland Lam, Ricardo SIGNES, Moritz Onken.

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

Dist::Zilla::Plugin::Bugtracker - Automatically sets the bugtracker URL

VERSION

version 1.101490

SYNOPSIS

In dist.ini:

```
[Bugtracker]
```

DESCRIPTION

This plugin sets the distribution's bugtracker URL as metadata. This plugin assumes you are using the CPAN RT bugtracker and sets those 2 fields:

```
web = http://rt.cpan.org/Public/Dist/Display.html?Name=$dist
mailto = bug-$dist at rt.cpan.org
```

Please see CPAN::Meta::Spec for more information.

FUNCTIONS

metadata

Sets the bugtracker URL in the distribution's metadata.

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-Bugtracker/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-Bugtracker/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::HasVersionTests - Release tests for version numbers

VERSION

version 1.101420

SYNOPSIS

In dist.ini:

```
[HasVersionTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following file:

```
xt/release/has-version.t - a standard Test::HasVersion test
```

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-HasVersionTests>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-HasVersionTests/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-HasVersionTests/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::Git::Commit - commit dirty files

VERSION

version 1.102020

SYNOPSIS

In your dist.ini:

```
[Git::Commit]
changelog = Changes      ; this is the default
```

DESCRIPTION

Once the release is done, this plugin will record this fact in git by committing changelog and dist.ini. The commit message will be taken from the changelog for this release. It will include lines between the current version and timestamp and the next non-indented line.

The plugin accepts the following options:

- * `changelog` - the name of your changelog file. defaults to `Changes`.
- * `allow_dirty` - a file that will be checked in if it is locally modified. This option may appear multiple times. The default list is `dist.ini` and the changelog file given by `changelog`.
- * `commit_msg` - the commit message to use. defaults to `v%v%n%n%c`, meaning the version number and the list of changes.

You can use the following codes in `commit_msg`:

`%c`

The list of changes in the just-released version (read from changelog).

`%{dd-MMM-yyyy}d`

The current date. You can use any CLDR format supported by `DateTime` . A bare `%d` means `%{dd-MMM-yyyy}d`.

`%n`

a newline

`%N`

the distribution name

`%v`

the distribution version

METHODS

get_commit_message

This method returns the commit message. The default implementation reads the Changes file to get the list of changes in the just-released version.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::BumpVersion - (DEPRECATED) bump the version number by one before building

VERSION

version 4.102345

SYNOPSIS

WARNING This plugin is deprecated and will be removed. It is generally useless. It does not do what you think it does.

If loaded, this plugin will ensure that the distribution's version number is bumped up by one (in the smallest already-defined version units) before building begins. In other words, if dist.ini's version reads 0.002 then the newly built dist will be 0.003.

AUTHOR

Ricardo SIGNES <tjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::UploadToCPAN - upload the dist to CPAN

VERSION

version 4.102345

SYNOPSIS

If loaded, this plugin will allow the release command to upload to the CPAN.

DESCRIPTION

This plugin looks for configuration in your dist.ini or (more likely) ~/.dzil/config.ini:

```
[%PAUSE]
username = YOUR-PAUSE-ID
password = YOUR-PAUSE-PASSWORD
```

If this configuration does not exist, it can read the configuration from ~/.pause, in the same format that cpan-upload requires:

```
user YOUR-PAUSE-ID
password YOUR-PAUSE-PASSWORD
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::CheckExtraTests

- check xt tests before release

VERSION

version 0.004

SYNOPSIS

In your dist.ini:

```
[CheckExtraTests]
```

DESCRIPTION

Runs all xt tests before release. Dies if any fail. Sets RELEASE_TESTING, but not AUTHOR_TESTING.

SEE ALSO

- Dist::Zilla

AUTHOR

David Golden <dagolden@cpan.org>

COPYRIGHT AND LICENSE

This software is Copyright (c) 2010 by David Golden.

This is free software, licensed under:

The Apache License, Version 2.0, January 2004

Dist::Zilla::Plugin::ShareDir - install a directory's contents as "ShareDir" content

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[ShareDir]
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Signature - sign releases with Module::Signature

VERSION

version 1.100930

DESCRIPTION

This plugin will sign a distribution using Module::Signature.

This plugin should appear after any other AfterBuild plugin in your dist.ini file to ensure that no files are modified after it has been run

ATTRIBUTES

sign

A string value. If archive then a signature will be created when an archive is being created. If always then the directory will be signed whenever it is built. Default is archive

This attribute can be overridden by an environment variable DZSIGN

AUTHOR

Graham Barr <gbarr@pobox.com>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Graham Barr.

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

Dist::Zilla::Plugin::PodSyntaxTests

- a release test for Pod syntax

VERSION

version 4.102345

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

xt/release/pod-syntax.t - a standard Test::Pod test

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::NoTabsTests - Release tests making sure hard tabs aren't used

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

- xt/release/no-tabs.t

a standard Test::NoTabs test

AUTHOR

Florian Ragwitz <rafl@debian.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MakeMaker - build a Makefile.PL that uses ExtUtils::MakeMaker

VERSION

version 4.102345

DESCRIPTION

This plugin will produce an ExtUtils::MakeMaker -powered Makefile.PL for the distribution. If loaded, the Manifest plugin should also be loaded.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Manifest - build a MANIFEST file

VERSION

version 4.102345

DESCRIPTION

If included, this plugin will produce a MANIFEST file for the distribution, listing all of the files it contains. For obvious reasons, it should be included as close to last as possible.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git::Check - check your git repository before releasing

VERSION

version 1.102020

SYNOPSIS

In your dist.ini:

```
[Git::Check]
allow_dirty = dist.ini
allow_dirty = README
changelog = Changes      ; this is the default
```

DESCRIPTION

This plugin checks that git is in a clean state before releasing. The following checks are performed before releasing:

- * there should be no files in the index (staged copy)
- * there should be no untracked files in the working copy
- * the working copy should be clean. The files listed in `allow_dirty` can be modified locally, though.

If those conditions are not met, the plugin will die, and the release will thus be aborted. This lets you fix the problems before continuing.

The plugin accepts the following options:

- * `changelog` - the name of your changelog file. defaults to `Changes`.
- * `allow_dirty` - a file that is allowed to have local modifications. This option may appear multiple times. The default list is `dist.ini` and the changelog file given by `changelog`. You can use `allow_dirty =` to prohibit all local modifications.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::Readme - build a README file

VERSION

version 4.102345

DESCRIPTION

This plugin adds a very simple README file to the distribution, citing the dist's name, version, abstract, and license. It may be more useful or informative in the future.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaYAML - produce a META.yml

VERSION

version 4.102345

DESCRIPTION

This plugin will add a META.yml file to the distribution.

For more information on this file, see Module::Build::API and CPAN::Meta .

ATTRIBUTES

filename

If given, parameter allows you to specify an alternate name for the generated file. It defaults, of course, to META.yml.

version

This parameter lets you pick what version of the spec to use when generating the output. It defaults to 1.4, the most commonly supported version at present.

This may change without notice in the future.

Once version 2 of the META file spec is more widely supported, this may default to 2.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git::Push - push current branch

VERSION

version 1.102020

SYNOPSIS

In your dist.ini:

```
[Git::Push]
push_to = origin          ; this is the default
push_to = origin HEAD:refs/heads/released ; also push to released branch
```

DESCRIPTION

Once the release is done, this plugin will push current git branch to remote end, with the associated tags.

The plugin accepts the following options:

- `push_to` - the name of the a remote to push to. The default is `origin`. This may be specified multiple times to push to multiple repositories.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::EOLTests - Release tests making sure correct line endings are used

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

- xt/release/eol.t

a standard Test::EOL test

ATTRIBUTES

trailing_whitespace

If this option is set to a true value, { trailing_whitespace => 1 } will be passed to all_perl_files_ok in Test::EOL . It defaults to 1.

AUTHOR

Florian Ragwitz <rafl@debian.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::AutoPrereqs - automatically extract prereqs from your modules

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[AutoPrereqs]
skip = ^Foo|Bar$
```

DESCRIPTION

This plugin will extract loosely your distribution prerequisites from your files using Perl::PrereqScanner .

If some prereqs are not found, you can still add them manually with the Dist::Zilla::Plugin::Prereqs plugin.

This plugin will skip the modules shipped within your dist.

ATTRIBUTES

skip

This string will be used as a regular expression. Any module names matching this regex will not be registered as prerequisites.

CREDITS

This plugin was originally contributed by Jerome Quelin.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::GatherDir - gather all the files in a directory

VERSION

version 4.102345

DESCRIPTION

This is a very, very simple FileGatherer plugin. It looks in the directory named in the root attribute and adds all the files it finds there. If the root begins with a tilde, the tilde is replaced with the current user's home directory according to File::HomeDir .

Almost every dist will be built with one GatherDir plugin, since it's the easiest way to get files from disk into your dist. Most users just need:

```
[GatherDir]
```

...and this will pick up all the files from the current directory into the dist. You can use it multiple times, as you can any other plugin, by providing a plugin name. For example, if you want to include external specification files into a subdir of your dist, you might write:

```
[GatherDir]
```

```
; this plugin needs no config and gathers most of your files
```

```
[GatherDir / SpecFiles]
```

```
; this plugin gets all the files in the root dir and adds them under ./spec
root   = ~/projects/my-project/spec
prefix = spec
```

ATTRIBUTES

root

This is the directory in which to look for files. If not given, it defaults to the dist root -- generally, the place where your dist.ini or other configuration file is located.

prefix

This parameter can be set to gather all the files found under a common directory. See the description above for an example.

include_dotfiles

By default, files will not be included if they begin with a dot. This goes both for files and for directories relative to the root.

In almost all cases, the default value (false) is correct.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::ReadmeFromPod - Automatically convert POD to a README for Dist::Zilla

SYNOPSIS

```
# dist.ini  
[ReadmeFromPod]
```

DESCRIPTION

Generates a plain-text README for your Dist::Zilla powered dist from its main_module with Pod::Text .

AUTHORS

Fayland Lam <fayland@gmail.com> and Ævar Arnfjörð Bjarmason <avar@cpan.org>

LICENSE AND COPYRIGHT

Copyright 2010 Fayland Lam <fayland@gmail.com> and Ævar Arnfjörð Bjarmason <avar@cpan.org>

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

Dist::Zilla::Plugin::AutoVersion - take care of numbering versions so you don't have to VERSION

version 4.102345

DESCRIPTION

This plugin automatically produces a version string, generally based on the current time. By default, it will be in the format: 1.yyDDdn

ATTRIBUTES

major

The major attribute is just an integer that is meant to store the major version number. If no value is specified in configuration, it will default to 1.

This attribute's value can be referred to in the autoversion format template.

format

The format is a Text::Template string that will be rendered to form the version. It is meant to access to one variable, \$major, and one subroutine, cldr, which will format the current time (in GMT) using CLDR patterns (for which consult the DateTime documentation).

The default value is:

```
{{ $major }}.{{ cldr('yyDDD') }}
{{ sprintf('%01u', ($ENV{N} || 0)) }}
{{ $ENV{DEV} ? (sprintf '_%03u', $ENV{DEV}) : '' }}
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaNoIndex - Stop CPAN from indexing stuff

VERSION

version 1.101130

SYNOPSIS

In your dist.ini:

```
[MetaNoIndex]
directory = t/author
directory = examples
file = lib/Foo.pm
package = My::Module
namespace = My::Module
```

DESCRIPTION

This plugin allows you to prevent PAUSE/CPAN from indexing files you don't want indexed. This is useful if you build test classes or example classes that are used for those purposes only, and are not part of the distribution. It does this by adding a `no_index` block to your `META.yml` file in your distribution.

ATTRIBUTES

directory

Exclude folders and everything in them. Example: `author.t`. Aliases: `folder`, `dir`, `directories`.

file

Exclude a specific file. Example: `lib/Foo.pm`. Alias: `files`.

package

Exclude by package name. Example: `My::Package`. Aliases: `class`, `module`, `packages`.

namespace

Exclude everything under a specific namespace. Example: `My::Package`. Alias: `namespaces`.

NOTE: This will not exclude the package `My::Package`, only everything under it like `My::Package::Foo`.

METHODS

metadata

Returns a reference to a hash containing the distribution's `no_index` metadata.

. **SEE ALSO**

AUTHORS

Mark Gardner <MJGARDNER@cpan.org>

JT Smith <RIZEN@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Mark Gardner.

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

Dist::Zilla::Plugin::CriticTests - tests to check your code against best practices

VERSION

version 1.100140

SYNOPSIS

In your dist.ini:

```
[CriticTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

* t/author/critic.t - a standard test to check your code against best practices

This plugin does not accept any option yet.

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::CheckChangesHasContent - Ensure Changes has content before releasing

VERSION

version 0.003

SYNOPSIS

```
# in dist.ini
```

```
[CheckChangesHasContent]
```

DESCRIPTION

This is a "before release" Dist::Zilla plugin that ensures that your Changes file actually has some content since the last release. If it doesn't find any, it will abort the release process.

The algorithm is very naive. It looks for an unindented line starting with the version to be released. It then looks for any text from that line until the next unindented line (or the end of the file), ignoring whitespace.

For example, in the file below, algorithm will find "- blah blah blah":

```
Changes file for Foo-Bar
```

```
{{ $NEXT }}
```

```
- blah blah blah
```

```
0.001 Wed May 12 13:49:13 EDT 2010
```

```
- the first release
```

If you had nothing but whitespace between `{{ $NEXT }}` and 0.001, the release would be halted.

If you name your change log something other than "Changes", you can configure the name with the `changelog` argument:

```
[CheckChangesHasContent]
changelog = ChangeLog
```

SEE ALSO

- Dist::Zilla

AUTHOR

David Golden <dagolden@cpan.org>

COPYRIGHT AND LICENSE

This software is Copyright (c) 2010 by David Golden.

Dist:Zilla

DESCRIPTION

This is free software, licensed under:

The Apache License, Version 2.0, January 2004

Dist:Zilla

110

Dist::Zilla::Plugin::InlineFiles - files in a data section

VERSION

version 4.102345

DESCRIPTION

This plugin exists only to be extended, and gathers all files contained in its data section and those of its ancestors. For more information, see `Data::Section`.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::PodCoverageTests - a release test for Pod coverage

VERSION

version 4.102345

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

```
xt/release/pod-coverage.t - a standard Test::Pod::Coverage test
```

This test uses Pod::Coverage::TrustPod to check your Pod coverage. This means that to indicate that some subs should be treated as covered, even if no documentation can be found, you can add:

```
=for Pod::Coverage sub_name other_sub this_one_too
```

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::Git - update your git repository after release

VERSION

version 1.102020

DESCRIPTION

This set of plugins for Dist::Zilla can do interesting things for module authors using git to track their work.

- The following plugins are provided in this distribution:

A Dist::Zilla::PluginBundle::Git bundle is also provided for your convenience, check its documentation to see which plugins it will load.

SEE ALSO

You can look for information on this module at:

- * Search CPAN

<http://search.cpan.org/dist/Dist-Zilla-Plugin-Git>

- * See open / report bugs

<http://rt.cpan.org/NoAuth/Bugs.html?Dist=Dist-Zilla-Plugin-Git>

- * Mailing-list (same as Dist::Zilla)

http://www.listbox.com/subscribe/?list_id=139292

- * Git repository

<http://github.com/jquelin/dist-zilla-plugin-git>

- * AnnoCPAN: Annotated CPAN documentation

<http://annocpan.org/dist/Dist-Zilla-Plugin-Git>

- * CPAN Ratings

<http://cpanratings.perl.org/d/Dist-Zilla-Plugin-Git>

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::Prepender - prepend lines at the top of your perl files

VERSION

version 1.100960

SYNOPSIS

In your dist.ini:

```
[Prepender]
copyright = 0
line = use strict;
line = use warnings;
```

DESCRIPTION

This plugin will prepend the specified lines in each Perl module or program within the distribution. For scripts having a shebang line, lines will be inserted just after it.

This is useful to enforce a set of pragmas to your files (since pragmas are lexical, they will be active for the whole file), or to add some copyright comments, as the fsf recommends.

The module accepts the following options in its dist.ini section:

* copyright - whether to insert a boilerplate copyright comment. defaults to true.

* line - anything you want to add. may be specified multiple times. no default.

SEE ALSO

You can look for information on this module at:

* Search CPAN

<http://search.cpan.org/dist/Dist-Zilla-Plugin-Prepender>

* See open / report bugs

<http://rt.cpan.org/NoAuth/Bugs.html?Dist=Dist-Zilla-Plugin-Prepender>

* Mailing-list (same as Dist::Zilla)

http://www.listbox.com/subscribe/?list_id=139292

* Git repository

<http://github.com/jquelin/dist-zilla-plugin-prepend>

* AnnoCPAN: Annotated CPAN documentation

<http://annocpan.org/dist/Dist-Zilla-Plugin-Prepender>

* CPAN Ratings

<http://cpanratings.perl.org/d/Dist-Zilla-Plugin-Prepender>

AUTHOR

Jerome Quelin

COPYRIGHT AND LICENSE

This software is copyright (c) 2009 by Jerome Quelin.

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

Dist::Zilla::Plugin::ManifestSkip - decline to build files that appear in a MANIFEST.SKIP-like file

VERSION

version 4.102345

DESCRIPTION

This plugin reads a MANIFEST.SKIP-like file, as used by ExtUtils::MakeMaker and ExtUtils::Manifest , and prunes any files that it declares should be skipped.

ATTRIBUTES

skipfile

This is the name of the file to read for MANIFEST.SKIP-like content. It defaults, unsurprisingly, to MANIFEST.SKIP.

AUTHOR

Ricardo SIGNES <rijbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::AutoPrereq - (DEPRECATED) the old name for Dist::Zilla::Plugin::AutoPrereqs VERSION

version 4.102345

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::GenerateFile - build a custom file from only the plugin configuration

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[GenerateFile]
filename      = todo/master-plan.txt
is_template  = 1
content      = # Outlines the plan for world domination by {{$dist->name}}
content      =
content      = Item 1: Think of an idea!
content      = Item 2: ?
content      = Item 3: Profit!
```

DESCRIPTION

This plugin adds a file to the distribution.

You can specify the content, as a sequence of lines, in your configuration. The specified content might be literal, or might be a Text::Template template.

Templating of the content

If you provide a `is_template` parameter of "1", The content will also be run through Text::Template. The variables `$plugin` and `$dist` will be provided, set to the GenerateFile plugin and the Dist::Zilla object respectively.

ATTRIBUTES

filename

This attribute names the file you want to generate. It is required.

content

The content attribute is an arrayref of lines that will be joined together with newlines to form the file content.

is_template

This attribute is a bool indicating whether or not the content should be treated as a Text::Template template. By default, it is false.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::ExecDir - install a directory's contents as executables

VERSION

version 4.102345

SYNOPSIS

In your dist.ini:

```
[ExecDir]  
dir = scripts
```

If no dir is provided, the default is bin.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::FakeRelease

- fake plugin to test release

VERSION

version 4.102345

SYNOPSIS

```
[FakeRelease]  
user = CPANAUTHORID ; # optional.
```

DESCRIPTION

This plugin is a Releaser that does nothing. It is directed to plugin authors, who may need a dumb release plugin to test their shiny plugin implementing BeforeRelease and AfterRelease.

When this plugin does the release, it will just log a message and finish.

If you set the environment variable DZIL_FAKERELEASE_FAIL to a true value, the plugin will die instead of doing nothing. This can be useful for authors wanting to test reliably that release failed.

You can optionally provide the 'user' parameter, which defaults to 'AUTHORID', which will allow things that depend on this metadata (Sometimes provided by Dist::Zilla::Plugin::UploadToCPAN) to still work. (For example: Dist::Zilla::Plugin::Twitter)

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::MetaJSON - produce a META.json

VERSION

version 4.102345

DESCRIPTION

This plugin will add a META.json file to the distribution.

This file is meant to replace the old-style META.yml. For more information on this file, see Module::Build::API and CPAN::Meta .

ATTRIBUTES

filename

If given, parameter allows you to specify an alternate name for the generated file. It defaults, of course, to META.json.

version

This parameter lets you pick what version of the spec to use when generating the output. It defaults to 2 at present, but may be updated to new specs as they are released and adopted.

If you want a fixed version, specify it.

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Dist::Zilla::Plugin::KwaliteeTests

- Release tests for kwalitee

VERSION

version 1.101420

SYNOPSIS

In dist.ini:

```
[KwaliteeTests]
```

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following file:

```
xt/release/kwalitee.t - a standard Test::Kwalitee test
```

INSTALLATION

See perlmodinstall for information and options on installing Perl modules.

BUGS AND LIMITATIONS

No bugs have been reported.

Please report any bugs or feature requests through the web interface at <http://rt.cpan.org/Public/Dist/Display.html?Name=Dist-Zilla-Plugin-KwaliteeTests>.

AVAILABILITY

The latest version of this module is available from the Comprehensive Perl Archive Network (CPAN). Visit <http://www.perl.com/CPAN/> to find a CPAN site near you, or see <http://search.cpan.org/dist/Dist-Zilla-Plugin-KwaliteeTests/>.

The development version lives at <http://github.com/hanekomu/Dist-Zilla-Plugin-KwaliteeTests/>. Instead of sending patches, please fork this project using the standard git and github infrastructure.

AUTHOR

Marcel Gruenauer <marcel@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Marcel Gruenauer.

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

Dist::Zilla::Plugin::MetaProvides::FromFile
- In the event nothing else works, pull in hand-crafted metadata from a specified file.

VERSION

version 1.10027802

ROLES

Dist::Zilla::Role::MetaProvider::Provider

PLUGIN FIELDS

file

type: required, ro, Str

reader_name

type: ClassName, ro.

default: Config::INI::Reader

PRIVATE PLUGIN FIELDS

_reader

type: Object, ro, built from reader_name

ROLE SATISFYING METHODS

provides

A conformant function to the Dist::Zilla::Role::MetaProvider::Provider Role.

signature: \$plugin->provides()

returns: Array of

Dist::Zilla::MetaProvides::ProvideRecord

BUILDER METHODS

_build_reader

. **SEE ALSO**

AUTHOR

Kent Fredric <kentnl@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 by Kent Fredric.

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

Dist::Zilla::Plugin::MetaTests - common extra tests for META.yml

VERSION

version 4.102345

DESCRIPTION

This is an extension of Dist::Zilla::Plugin::InlineFiles , providing the following files:

xt/release/meta-yaml.t - a standard Test::CPAN::Meta test

AUTHOR

Ricardo SIGNES <rjbs@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.

Appendix

COPYRIGHT AND LICENSE

This software is copyright (c) 2010 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.