



# Dist::Zilla

distribution builder; installer not included  
est. 11 Feb 2011

Presentation by Matthew B. Gray  
Actual authorship by Ricardo Signes

# The Overhead in CPANning

Given a file `Distribution-Name-1.234.tar.gz`

You (may) have to (on a good day) maintain...

`Distribution-Name-1.234/Changes`

`Distribution-Name-1.234/LICENSE`

`Distribution-Name-1.234/MANIFEST`

`Distribution-Name-1.234/META.yml`

`Distribution-Name-1.234/Makefile.PL`

`Distribution-Name-1.234/README`

`Distribution-Name-1.234/bin/dist-program`

`Distribution-Name-1.234/lib/Distribution/Name.pm`

`Distribution-Name-1.234/lib/Distribution/Name/Helper.pm`

`Distribution-Name-1.234/t/00-compile.t`

`Distribution-Name-1.234/t/blowfish-tests.t`

`Distribution-Name-1.234/t/monkfish-tests.t`

`Distribution-Name-1.234/t/octopode-tests.t`

`Distribution-Name-1.234/xt/release/pod-syntax.t`

# Yeah, so what? I like pain.

- All of the files in the distribution are managed by the author
- The author maintains the manifest (ew)
- They need to provide the correct license file (harder when dealing with many projects)
- And keep copy pasting those stock tests we want to ship with distributions (i.e. perl-critic, pod-coverage, pod-syntax)

*OMG, get me out of here!!*

**XML!**

# Just kidding

[Raw] Dist::Zilla, by default, provides the ability to upload distributions to CPAN with a single command (i.e. `dzil release`). Dist::Zilla also has functionality for creating new distributions (i.e. `dzil new My::New::Module`). It also automatically generates many of the files that [usually need to be] maintain by hand.

<http://stackoverflow.com/questions/3410108/what-are-the-strengths-weaknesses-of-shipit-vs-distzilla>

Everything else you'll ever want to do should be scaffolded by plugins

<http://search.cpan.org/search?query=dist%3A%3Azilla%3A%3A&mode=all>

Interestink maths:

2 Months, 418 extensions/modules ~ 200p/m ~ 46p/w

# Minting your distzilla setup

## **dzil setup**

What's your name? Matthew B. Gray

What's your email address? matthew.b.gray@gmail.com

Who, by default, holds the copyright on your code? [Matthew B. Gray]:

What license will you use by default (Perl\_5, BSD, etc.)? [Perl\_5]: GPL3

Invalid selection, please try again: [Perl\_5] GPL\_3

Do you want to enter your PAUSE account details? [y/N]: N

config.ini file created!

# Creating a new Distribution

```
mbgray@mantis:~ dzil new Russian::Guyo::Vich
```

```
[DZ] making target dir
```

```
/home/mbgray/workspace/distdilla/Russian-Guyo-Vich
```

```
[DZ] writing files to /home/mbgray/workspace/distdilla/Russian-Guyo-Vich
```

```
[DZ] dist minted in ./Russian-Guyo-Vich
```

What does that do?

Creates a folder Russian-Guyo-Vich

Gives you a basic setup...

```
./dist.ini
```

```
./lib
```

```
./lib/Russian
```

```
./lib/Russian/Guyo
```

```
./lib/Russian/Guyo/Vich.pm
```



# dist.ini, the generated kind

```
name    = Russian-Guyo-Vich  
author  = Matthew B. Gray  
license = GPL_3  
copyright_holder = Matthew B. Gray  
copyright_year  = 2011
```

```
version = 0.001
```

```
[@Basic]
```



# Plugin Arch, Expanding [@Basic]

This is how we describe Dist::Zilla groups of plugins. @Basic contains the following plugins

- [GatherDir]
- [MetaYAML]
- [MakeMaker]
- [Manifest]
- [PruneCruft]
- [License]
- [Readme]
- [ExtraTests]

When you edit your dist version, license, or abstract, your files will be guaranteed to contain the new data.

# Moar! Plugins -- the recommended kind

Some other useful pluggins that you should add to run stock tests.

[PodSyntaxTests]

[PodCoverageTests]

[CriticTests]



# Building a Project

The build occurs in phases, but these *can be tweaked*

What tends to happen...

1. Files are gathered into a big pile
2. some are removed
3. some are rewritten
4. stuff happens
5. then finally the files are all written to the build destination.

It is important to realise that a single plugin can perform many of these roles.

When more than one plugin can perform work in a given phase, the plugins are run in the order they were listed in configuration.

```
~/workspace/distdilla/Russian-Guyo-Vich$ dzil build
```

```
[DZ] beginning to build Russian-Guyo-Vich
```

```
[DZ] guessing dist's main_module is lib/Russian/Guyo/Vich.pm
```

```
[DZ] extracting distribution abstract from lib/Russian/Guyo/Vich.pm
```

```
Unable to extract an abstract from lib/Russian/Guyo/Vich.pm. Please  
add the following comment to the file with your abstract:
```

```
# ABSTRACT: turns baubles into trinkets
```

```
~/workspace/distdilla/Russian-Guyo-Vich$ dzil build
```

```
[DZ] beginning to build Russian-Guyo-Vich
```

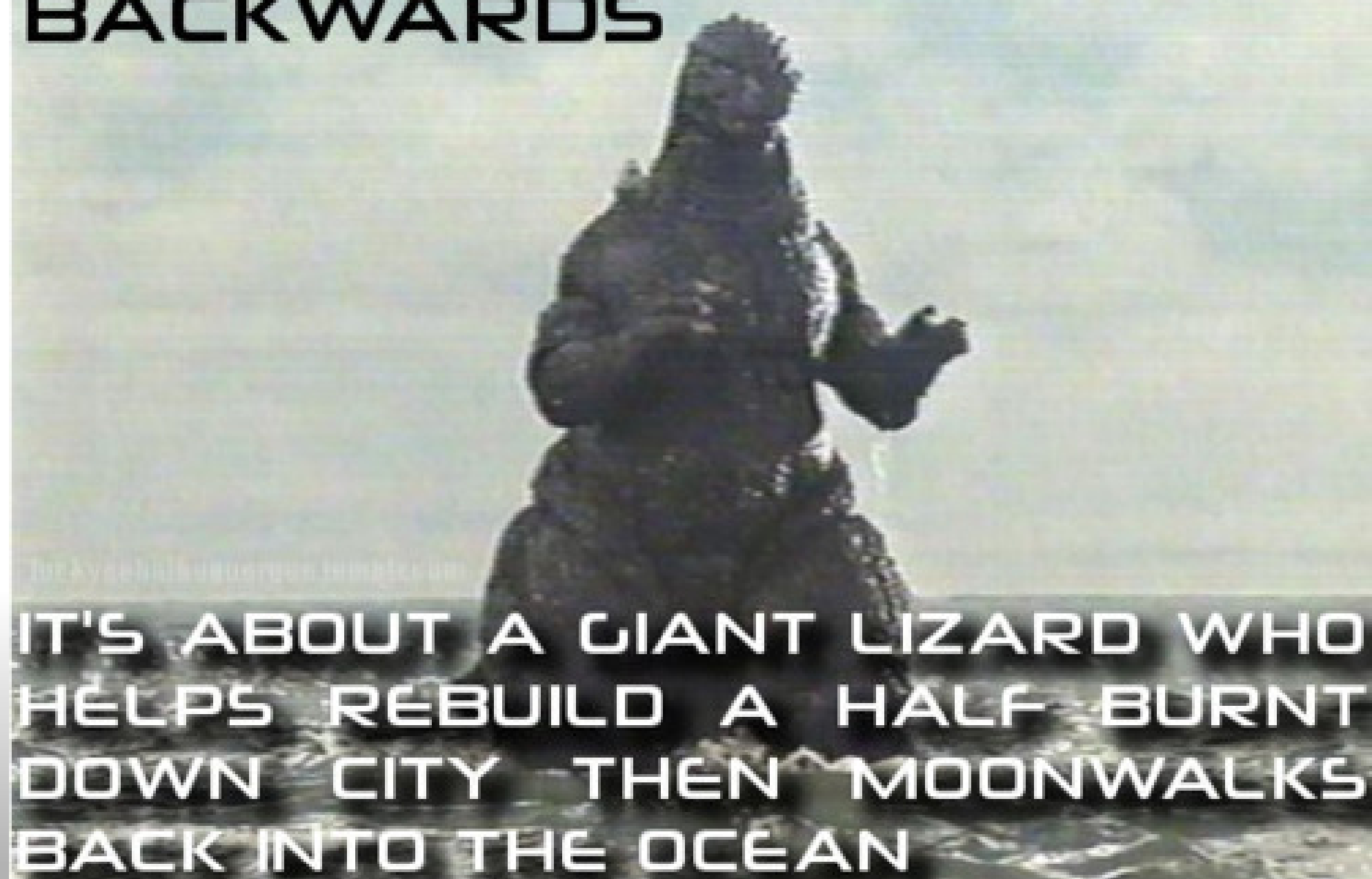
```
[DZ] guessing dist's main_module is lib/Russian/Guyo/Vich.pm
```

```
[DZ] extracting distribution abstract from lib/Russian/Guyo/Vich.pm
```

```
[DZ] writing Russian-Guyo-Vich in Russian-Guyo-Vich-0.001
```

```
[DZ] writing archive to Russian-Guyo-Vich-0.001.tar.gz
```

# IF YOU WATCH GODZILLA BACKWARDS

A large, dark, scaly Godzilla is walking on a beach, facing right. The background is a hazy, overcast sky and ocean. The text is overlaid on the image.

IT'S ABOUT A GIANT LIZARD WHO  
HELPS REBUILD A HALF-BURNT  
DOWN CITY THEN MOONWALKS  
BACK INTO THE OCEAN

# Testing

dzil test

... or for the impatient, prove -lr t

- Your files won't be autoversioned, if you depend on distzilla constructed files they won't be there.
- If your dist uses XS it won't work (but you're not wrapping, you're hacking in your spare time, right :)).

If you release before you test, someone will hit you in the face.

# Releasing to ze CPAN

Some helpful plugins that do stuff for you  
[TestRelease] #Don't leave home without it!  
[ConfirmRelease]  
[UploadToCPAN]

If you're making an official release, you can push with this command:  
dzil release

If you want to only release to testers, you can push like this:  
dzil release --trial

# Before you start, wouldn't do this one...

```
sudo apt-cache policy libdist-zilla-perl
```

```
libdist-zilla-perl:
```

```
  Installed: 4.101612-1
```

```
  Candidate: 4.101612-1
```

```
  Version table:
```

```
*** 4.101612-1 0
```

```
    500 http://nz.archive.ubuntu.com/ubuntu/ maverick/universe amd64
```

```
Packages
```

```
  100 /var/lib/dpkg/status
```

```
sudo apt-cache policy libdist-zilla-plugin-podweaver-perl
```

```
libdist-zilla-plugin-podweaver-perl:
```

```
  Installed: 3.101530-1
```

```
  Candidate: 3.101530-1
```

```
  Version table:
```

```
*** 3.101530-1 0
```

```
    500 http://nz.archive.ubuntu.com/ubuntu/ maverick/universe amd64
```

```
Packages
```

```
  100 /var/lib/dpkg/status
```



# What I did...

```
time sudo perl -MCPAN -e 'install Dist::Zilla'
```

...screens of junk later...

Warning (usually harmless): 'YAML' not installed, will not store persistent state

```
real 6m58.693s
```

```
user 4m48.840s
```

```
sys 0m27.310s
```

*hmmm?*

```
sudo apt-get install libyaml-perl
```