

Carton

Tatsuhiko Miyagawa
YAPC::Asia 2011 Tokyo

Managing CPAN Dependencies

Case Study: (Poor) Web Development

You're writing a new web app.
You don't wanna reinvent wheels.
Want to use as many CPAN deps.

Fine.
Get them from CPAN,
install on your machine.

```
> cpanm Web::Framework
installed LWP-5.912
installed Plack-0.9980
installed Web-Framework-1.20
> cpanm JSON::Fast
installed JSON-Fast-1.91
> cpanm MIME::Parser::XS
installed MIME-Parser-XS-0.20
```

Test it...

Works? Ship it!

Deployment

Set up new production servers.
Install CPAN modules
(until the errors are gone)

Few weeks later...

Your website is popular!
Need more web servers!

Re-Deployment

Set up new production servers.
Install CPAN modules
(until the errors are gone)

“Crap, Web::Framework has been updated to 1.4 and many APIs have been changed or deprecated!”

Rollback

Log in to the old web server.
Check Perl module versions.
Install them on the new server.

“Crap, this author deleted
the version 1.20 we want.
Let’s go to BackPAN...”

“Crap, version 1.20 doesn’t actually work with the newer LWP 6 that we just installed! Have to downgrade this too..”

and so on.

What was wrong?

- Dependency declaration
- Isolated Perl environments
- Version controls/history
- Dependency analysis
- Repeatable deployments
- etc.

Many existing solutions

MyCPAN, DPAN, CPAN::Mini::Inject,
OrePAN, Shipwright

None of those
didn't quite work for me.

(Or I haven't even tried)

So I wrote a new one.

Carton

<https://github.com/miyagawa/carton>

Inspired by...



The best way to manage your
application's dependencies

Bundler

Bundler manages an **application's dependencies** through its entire life across many machines systematically and repeatably.

I am interested in

[Troubleshooting](#)[Understanding Bundler](#)[Gemfile Manual](#)[CLI Manual](#)

Getting Started

Getting started with bundler is easy

```
$ gem install bundler
```

- App-specific local environment
- Fast and safe install
- Dep-tree analysis, including versions
- Locking module versions
- Easy Redeployment
- Single-file, VCS friendly
- Safe and easy rollback

Local perl environment

Using local::lib and cpanm -L
Each app has an isolated local library path

Fast and safe install

`cpanm 1.5`

Saves MYMETA.json and install meta info

Dep tree analysis

Rebuild the dependency tree from meta info
Checks if anything is missing/superfluous

Locking versions

Versions are saved in carton.lock
including dependencies

Easy Redeployment

Reinstall exactly the same set of modules
on another prod/development machines.

Single-file, VCS friendly

You can add carton.lock to git
update whenever you update modules

Safe and easy rollback

revert the lock file and redeploy

DEMO

> cpanm Carton

WARNING

It is beta software, some features are missing or not working correctly (yet).

github.com/miyagawa/carton
[#carton](https://irc.perl.org)

Questions?

Thanks!

twitter.com/miyagawa