On The Menu Tonight...

- Reasons To Create CPAN Distros
- Case Study: Foo Financial Company
- Assorted Tools You Can Use
This presentation was originally titled...

Why You Should Create CPAN Distros
Why Perl Applications Should Have Build Systems

...and CPAN is just one way to do it.
Well, why?
Organization!

- Products
  - Distributions
    - Modules
      - Subroutines
        - Statements
A Well-Known Layout

A proper place for...

- scripts
- libraries
- tests
Dependency Management

* CPAN.pm Does All The Work
* Meta-Version Control
* Detect Circular Dependencies
* Identify Dependency Flavors:
  * “requires”
  * “build_requires”
  * “configure_requires”
  * “author_requires”
Separation Of Concerns

Author-time → Build-time → Install-time

☆ Usernames and Passwords
☆ Resource Paths
☆ Configuration Stuff
Sharing And Distribution

- Easy to identify (i.e. named, versioned)
- Easy to transport (as *.tar.gz files)
- Well-known install procedure
Tool Chain Support

- CPAN
- Module::Build
- More on this later...
Case Study: Foo Financial Company

- 500k lines of Perl code
- 100s of scripts, 100s of modules
- No automated test cases
- RCS-like version control
- Quasi-separate DEV and PROD
- Deploy by manually copying to PROD
Havoc Ensued

Which files need to be deployed?
Which versions of the files?
How do I role back?
What are the parts of this system?
How do changes affect other parts?
A Solution:

* Divide product into Distros.
* Publish Distros into a private CPAN.
* Assemble product from private CPAN.
* Automate everything!
SVN

trunk/
  Foo-Reporting/
  Foo-Interfaces/
  Foo-Utilities/
  Foo-Everything/

trunk/CPAN/authors/id/F/FOO/
  Foo-Reporting-8.01.tar.gz
  Foo-Interfaces-2.37.tar.gz
  Foo-Utilities-6.01.tar.gz
  Foo-Everything-1.02.tar.gz

CruiseControl

$ ./Build
$ ./Build test
$ ./Build install
$ ./Build dist

add & commit

HTTP GET

$ cpan Foo-Everything

installs

Foo-Reporting-8.01.tar.gz

archives

Foo-reporting-8.01.tar.gz

Finished product!

Foo-1.45.tar.gz

A temporary location, that will contain all Foo::* modules and their non-core dependencies

A private CPAN, containing everything that Foo depends on

Empty Distro. Simply depends on other top-level Foo::* modules

Every commit produces a Distro with a new version number

Every commit produces a Distro with a new version number
Some Tips:

* At first, put Everything in one fat Distro.
* But later, may 1000 Distros bloom!
* Think about how to name & number versions.
* Create special accounts for automated jobs.
* Any SCM and build server will do the job.
* Know your target Perl environment.
Some Supporting Tools

- CPAN::Mini
- CPAN::Mini::Inject
- CPAN::Site
- Module::Starter
- Dist::Zilla
Jeffrey Thalhammer

IMAGINATIVE
Software Systems

Perl:::
Critic

jeff@imaginative-software.com