

Perl

Post Install Tests

April 11, 2013

Upgrading dependencies w/fear

1. Install distribution containing a module named `Bear`
2. Install distribution containing module `Human`, which depends on the availability of `Bear`'s `ride()` method
3. (time passes...)
4. Upgrade to a newer distribution version of `Bear`, which does NOT have the `ride()` method; The author of `Bear` heard it's not safe to ride bears.
5. ***BOOM*** `Human`'s attempt to `ride()` fails!
6. `Human` gets eaten by `Bear` and/or customer.

Sources of *upgrade fear*

- ▶ Don't assume CPAN authors have a commitment to their APIs. They may change at any time - even if the CPAN/Perl community has a history of preserving backwards compatibility!
- ▶ No *trivial* way of verifying if the installed Perl/CPAN dists still work.
- ▶ Modules might be used in ways unintended by the author.

Detecting breakage

We want to detect the upgrade bug/bear early to avoid getting eaten!
Our options are:

- ▶ Run your code and see if something breaks
- ▶ Run your code's integration/smoke tests
- ▶ Check cpantesters.org

We want more options!

- ▶ Run tests for all modules/distributions found in @INC
- ▶ Run tests for the dists that depend on the one you're about to upgrade
- ▶ Run tests in any system environment (development, test/CI, staging and if necessary, **production**)

Post Install Test requirements

- ▶ The tests for the installed modules must be available (installed).
- ▶ It must be possible to locate tests for a specific distribution and version.
- ▶ Distribution dependencies are saved and installed.
- ▶ It must be possible to determine a distribution's reverse dependencies, so we can only run the tests that are affected by a specific dist upgrade.
- ▶ Writing a *best practices* guide on how to make tests that can be used post-install.

Demo - Module::Build::PIT

- ▶ Extension of Module::Build.
- ▶ Environment variable PERL_INSTALL_TESTS controls if tests should be installed when using `./Build install` (Also possible to use `./Build installtests` for the same effect w/o ENV).
- ▶ Tests are installed under `$install_base/auto/tests/$distname-$distversion/`.
- ▶ Test files are written to the `.packlist` file.
- ▶ Action `testinc` uses the `ExtUtils::Installed` distribution to retrieve the modules that would be loaded and their test files.
- ▶ Action `testrdeps` is a hack, and depends atm. on MetaCPAN for demonstration purposes.

Future Work / Considerations

- ▶ Permissions - Test that writes to devices, sockets(ports < 1024), t/... and more
- ▶ Currently no trivial way of finding reverse dependencies locally
- ▶ Integration with packaging systems
- ▶ Upgrading test_requires to regular requires?
- ▶ Best practices doc

Code

- ▶ Code and slides are on github!
- ▶ <https://github.com/bannaN/Module-Build-PIT>
- ▶ <https://github.com/bannaN/qa-hackathon-presentation>
- ▶ Fork it now!