Perl
Post Install Tests

April 10, 2013

Upgrading dependencies w\o fear

1. You install a module named `Bear`
2. You install the module `Human`, which depends on the availability of Bear's `ride()` method
3. You install an new version of `Bear`, which does NOT have the `ride()` method; it's not safe to ride a bear!
4. Human's attempt to `ride()` fails
5. `Human` gets eaten by `Bear`

Sources of upgrade fear

- We should not assume that CPAN authors have a formal commitment to their APIs, they may change at any time
- No trivial way of verifying the whole perl installation with distributions
- CPAN/Perl community has usually been good at preserving backwards compatibility
- Modules might get used in an unintended way from the authors perspective

How to detect

You could do the following and hope that you detect the bug\breakage:

- Run your code and see
- Run tests for YOUR code
- Check cpantesters.org if you are running cutting edge from CPAN

Goals

- Run tests for all distributions loadable in `@INC`
- Run tests for the distributions that depend on the newly upgraded distribution
- Run tests in all environments (development, test/CI, staging and **production**)

Post Install Tests Requirements

- The tests for the installed modules must be available(installed)
- It must be possible to locate tests for a specific distribution and version
- In order to run the tests for only the dependent distributions it must be possible to do determine this distribution's reverse dependencies
- Distribution dependencies are saved and installed
- Writing a best practices guide for tests

Demo

- 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` without the ENV)
- Tests is installed under `$install_base/auto/tests/$dname-$dversion/`
- 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 fake action. The logic is hardcoded 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
- Best practices doc