Regression Testing

Gavan Fantom

gavan@NetBSD.org

pkgsrcCon 2005

Introduction

- Have you ever committed anything to mk?
- Did you break stuff?
- Has anybody else broken your code?
- What testing do you do before you commit?

Regression Testing Framework

- Automated tests of pkgsrc infrastructure
- Designed at pkgsrcCon 2004
- Will solve all the world's problems

  (except those solved by pkgviews)
- But not a substitute for other forms of testing

How to run regression tests

- Install pkgtools/pkg_regress
- Run pkg_regress
  - pkg_regress -v shows more details
- Tests live in regress/

Why should you run regression tests?

- Make sure stuff is broken before you commit
- Notice breakage more quickly
- You should run regression tests more often if you use non-standard settings or an esoteric Operating System.

How to write a regression test

- Test a specific feature of the infrastructure
- A test contains:
  - spec file
  - Makefile (typically)
  - Any other files required
- A test is only a test if it contains a spec file
  - Other directories are ignored, so a test can consist of more than one package if necessary.

Regression test example

- regress/pkgfail
  - Makefile
  - spec
- Tests that PKG_FAIL_REASON does what it says on the tin

Makefile

```
DISTNAME=  regress-pkgfail-0.0
CATEGORIES=  regress
MAINTAINER=  gavan@NetBSD.org
COMMENT= Test PKG_FAIL_REASON

PKG_FAIL_REASON=  "This package should
  never build"

.include "../../mk/bsd.pkg.mk"
```

spec

```
MAKEARGS_TEST=install


check_result()
{
      exit_status 1
      output_require "This package should
       never build"
}
```

Things you can do in the spec file

- Override:
  - `do_setup, do_cleanup, do_test`
  - `check_result`
- Define:
  - `MAKEARGS_TEST`
  - `MAKEARGS_CLEAN`
- Use:
  - `exit_status status`
  - `output_require "Good Regular Expression"`
  - `output_prohibit "Bad Regular Expression"`

What makes a good test?

- Simplicity
- Platform-independence
- Environment-independence
- Consistency

What makes a bad test?

- Hard to understand
- Random or variable results
- Only works correctly on certain platforms
- Succeeds if infrastructure is broken

Why should you write tests?

- Stop people from breaking things you care about
- Formally specify desired pkgsrc behaviour
- Because you can
- Enter the competition for the most complicated regression test. Currently, jlam is winning.

Room for improvement

- Better reporting
- Locale support
- Support sub-tests
- Write more tests

Questions?

Regression Testing

Gavan Fantom

gavan@NetBSD.org

pkgsrcCon 2005