Spotting and Fixing Build Failures

Aaron M. Ucko
〈ucko@debian.org〉

August 20, 2015
Introduction

- ~ 600 FTBFS reports in 6 years?
Introduction

▶ ~ 600 FTBFS reports in 6 years?
▶ No official role needed, really, just a foreign chroot and an ad-hoc script.
▶ You can help too!
Introduction

▶ ~ 600 FTBFS reports in 6 years?
▶ No official role needed, really, just a foreign chroot and an ad-hoc script.
▶ You can help too!
▶ Focus on “NEW” binary packages – regressions of existing packages block migration. (Yes, I know that’s at the source package level, but don’t mind a somewhat broader scope.)
Common sources of trouble

- Build-Depends undeclared or misclassified.
- Unpreparedness for binary-only builds.
- Architecture-specific *symbols variation (mostly in C++ libs).
- Shortsighted upstream. (Yes, kFreeBSD and the Hurd exist!)
- DEB_HOST_GNU_TYPE vs. DEB_HOST_MULTIARCH (on *i386).
- Evolving build environment (depending on how long the package sat in NEW).
- Autobuilder quirks (no HOME, no network, etc.).
Checking on your own packages

  - Links to full logs.
  - Log tails for failed builds.
  - Specific errors sometimes preextracted.
- https://tracker.d.o/pkg/... (Subscribe button at top, supports SSO)
Checking on “NEW” packages

► https://people.debian.org/~ucko/compare-new-names
► Unpolished custom script (posted as is), taking advantage of aptitude’s tracking of new packages.
► On my system, it shows what packages are new on amd64 but unavailable altogether on i386, or vice versa.

Aaron M. Ucko ⟨ucko@debian.org⟩
Spotting and Fixing Build Failures
August 20, 2015
Checking on “NEW” packages

- [https://people.debian.org/~ucko/compare-new-names](https://people.debian.org/~ucko/compare-new-names)
- Unpolished custom script (posted as is), taking advantage of aptitude’s tracking of new packages.
- On my system, it shows what packages are new on amd64 but unavailable altogether on i386, or vice versa.
- `compare-new-names -q -u` yields a link to [https://buildd.d.o/status/package.php?...](https://buildd.d.o/status/package.php?...)
  - Uses compact mode as appropriate. (Icons: $\notin/\sim/\circ/\check\check/...$)
  - False positives are generally easy to pick out.
  - Can chase links to figure out why build deps are uninstallable.
- Check BTS and VCS before filing a new bug.
- Severity serious only for unrebuildability on release architectures.