

Removing obsolete packages for fun and profit

Eric Dorland eric@debian.org

DebConf 14, Portland


August 26, 2014

automake in Debian

- I started maintaining this around the automake 1.6 era. (First upload 11 Jun 2002)
- New releases of automake usually have some backwards compatibility problems.
- automake 1.5 broke tons of packages and had to be moved to its own package.
- Older versions get moved to their own package.
- The automake package tends to track the latest release, which is convenient.
- *Warning:* depending on automake directly can be risky
- Alternatives let you pick your `/usr/bin/automake`.

automake problems

- 1 Because of backwards compatibility issues, we package new versions in different packages.
- 2 Proliferation of automake packages in Debian: automake1.4, automake1.5, automake1.6, automake1.7, automake1.8, automake1.9, automake1.10, automake1.11, automake1.12 (only seen in experimental) automake-1.13 and automake-1.14.
- 3 At the wheezy release we had automake1.4, automake1.9, automake1.10 and automake (1.11)
- 4 automake1.4 was last released in 2002.
- 5 in sid we have automake-1.14 on top. 5 automake packages.



This is madness.

Stage 1: Denial

Denial

- “If I just send mail debian-devel asking for updates to a newer automake everyone will just do it, right?”

Denial

- “If I just send mail debian-devel asking for updates to a newer automake everyone will just do it, right?”
- Figure out which packages are at fault using `grep-dctrl` over the Sources file. (automake has only build dependents)
- 169 source packages.
- use `ddlist` to make this into a maintainer sorted list.
- Send email to debian-devel (May 27, 2013)

Stage 2: Anger

Anger

- “Crap, I’m going to actually have to do some work!”

Anger

- “Crap, I’m going to actually have to do some work!”
- Modified `lintian` to warn about the dependency via the `build-depends-on-obsolete-package` tag.
- Filed wishlist bugs with a simple script to just mail the BTS. Hurray email! (September 22, 2013)
- 117 bugs filed. 52 were fixed just by emailing `debian-devel` and waiting 4 months.
- Used the usertag `automake-cleanup-2013`

Stage 3: Bargaining

Bargaining

- “Look! Here’s a patch! You just have to apply it.”

Bargaining

- “Look! Here’s a patch! You just have to apply it.”
- 34 packages fixed without supplying a patch.
- Began in late October 2013.
- This is the hard part, tough to automate.
- Frequently just changing the build-deps was enough.
- Sometimes more elaborate patching of the build system was required.
- Used `pbuilder` to test for build success.
- Mailed it off to BTS and flipped the patch tag.

Stage 4: Depression

Depression

- “I guess I’m on my own here”

Depression

- “I guess I’m on my own here”
- Uploading NMUs
- Between January and April 2014 uploaded 63 NMUs to delayed upload
- Only 7 were overridden before the delay.
- Applied my patches from the BTS and created changelogs with debchange.
- Built with pbuilder and signed using gnupg-agent.
- Uploaded to DELAYED/10 and used nmudiff to update the bug.

Depression II

While many people thanked me, one maintainer wasn't happy:

```
> I've prepared an NMU for <redacted> and  
> uploaded it to DELAYED/10.  
NMUing for a wishlist bug???  
> Please feel free to tell me if I should delay it longer.  
Yes, please. indefinitely. (Or until is is at  
least important/RC)  
  
(author redacted to protect the guilty)
```

Stage 5: Acceptance

Acceptance

- “In the end it’s all in God’s^W FTP Master’s hands.”

Acceptance

- “In the end it’s all in God’s^W FTP Master’s hands.”
- automake1.4 and automake1.10 are gone!
- automake1.9 is still in the archive: three FTBFS packages still depend on it.
- Hope to sort that out here and get it out before the freeze.

Open Questions

- ❶ Should we just patch & NMU at the same time?
- ❷ What tools did I miss?
- ❸ What tools would make this easier?
- ❹ How can we go faster?
- ❺ Where is the fun and/or profit?