Building your own Haskell ecosystem

Haskell development using Hackage 2.0, cabal-dev and more.

Erik Hesselink

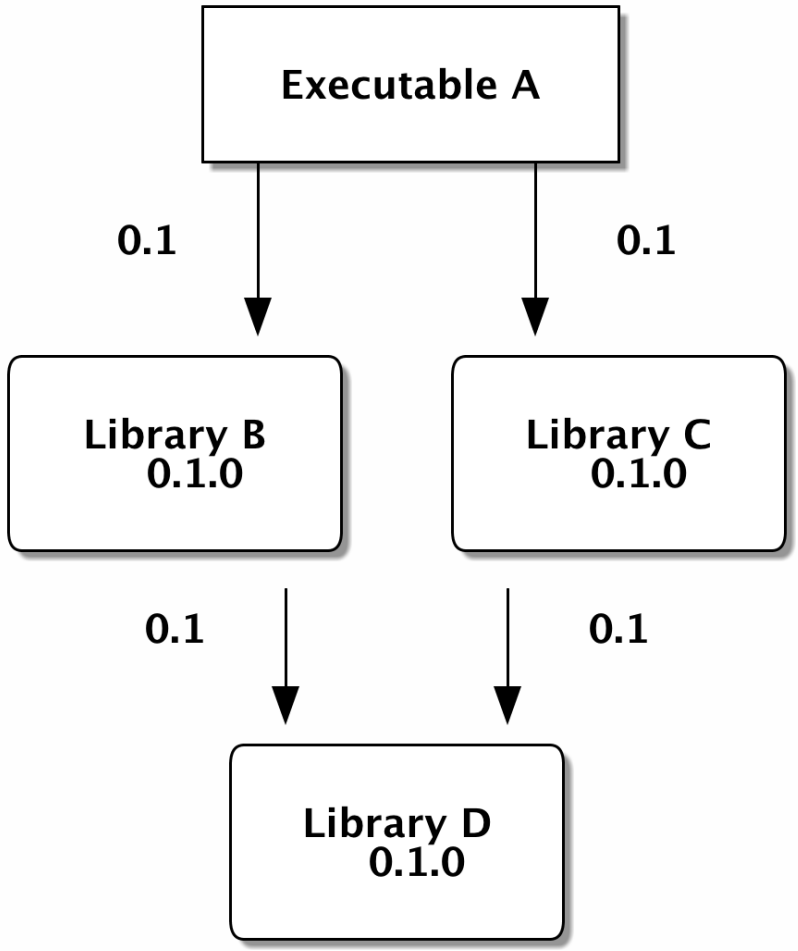
April 19, 2012

Context

- Programming Haskell.
- Multiple developers collaborating.
- Multiple inter-dependenent libraries.
- Closed source.

Problem

We have an executable, and three libraries:

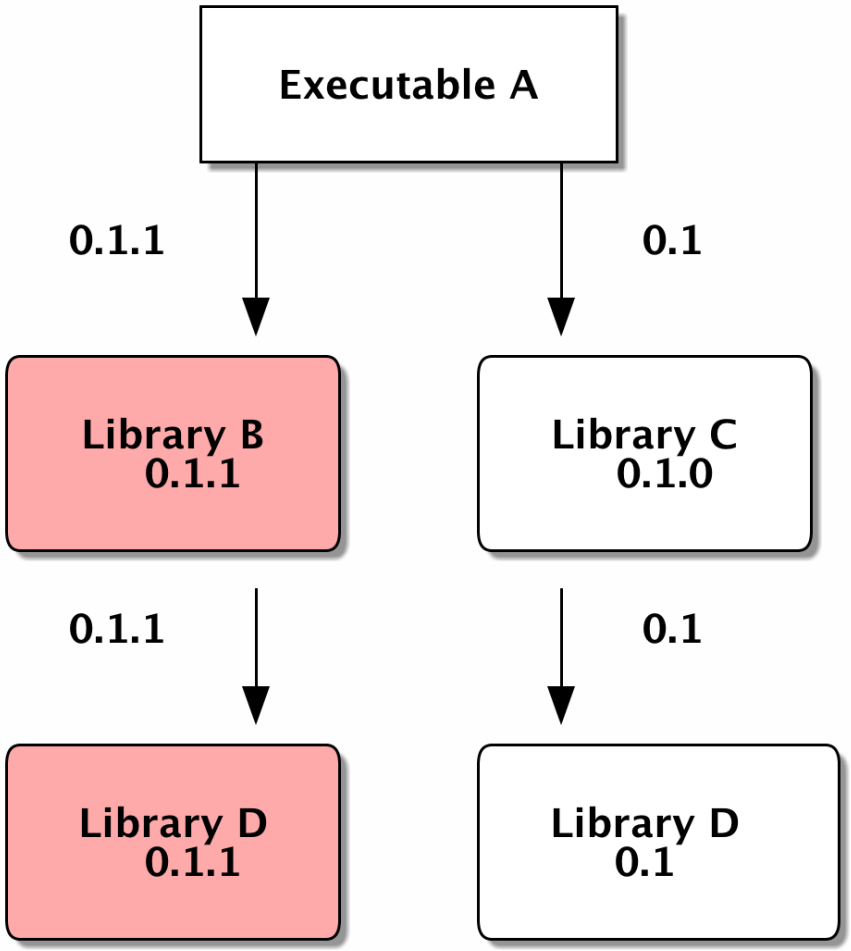


Problem - 2

- We develop a new feature in **D**, using it in **B**.
- Another developer pulls this in.
- What to do?

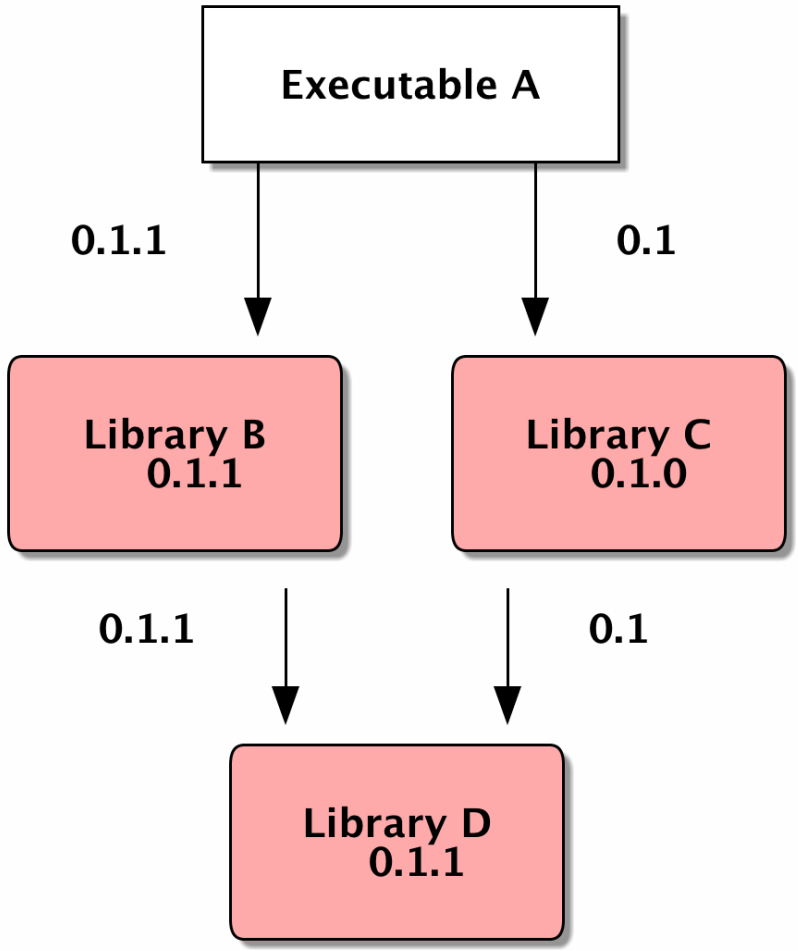
Problem - 3

Follow the errors: install **D**, then **B**. Failure:



Problem - 3

We want `cabal` to reinstall **C**, but it doesn't have sources.



Hackage 2.0

Solution: run your own Hackage!

⇒ Now `cabal` has the sources, and you can just run `cabal install` **A**.

```
darcs get http://code.haskell.org/hackage-server/
cd hackage-server
cabal install

hackage-server init --admin=NAME:PASS
hackage-server run --port=PORT
```

Access

- Hackage has public read access.
  - Not desirable for closed setting.
- Basic auth and https via `Apache` seems good.
  - But `cabal` supports neither.
- Now behind an IP whitelist.
  - Suboptimal, would like better solution.

Development

- Pulling (and deploying) finished code works.
- What about development?
- Developing deep in hierarchy still problematic.
- Multiple branches interfere with each other.

cabal-dev

`cabal-dev` provides a package sandbox.

- Wraps all cabal commands.
- Packages not installed in user db, but sandbox.
- Can use multiple sandboxes (`-s`).
- Can add source to sandbox.
- Available like disk-local hackage.

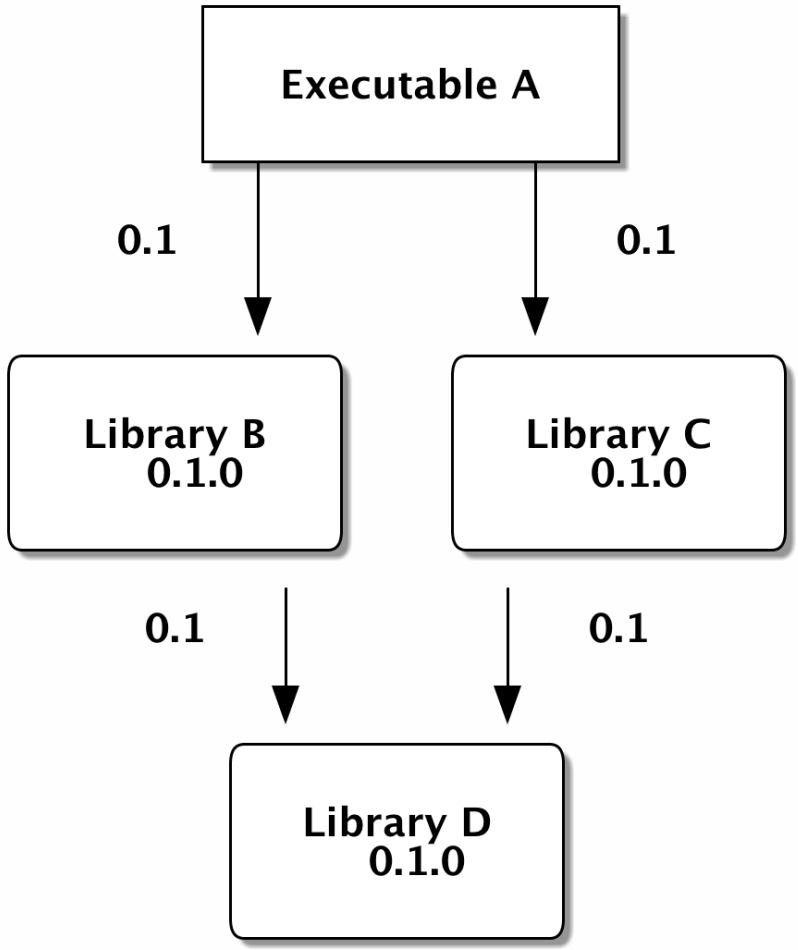
I heard you like wrappers

Remove all manual configuration.

- Wrapper for `cabal-dev` called `c`.
- Bases sandbox on current branch (and repo).
- Turns off costly things like profiling and documentation.
- Adds some commands, like `run`.
- Another script to add current source of all packages.

Dependencies reminder

Remember our executable and three libraries?



Development with c

```
# Hack on D (don't bump version).
c install
c add-source .
# Potentially repeat, change more libraries.
# Go to A.
c install
# Rebuilds all needed packages, like C.
c run A
```

Gotchas

- `ghci` doesn't work so well.
  - Needs `cabal repl`.
- Need manual sandbox cleanup.
  - Could use `git` hook.
- Full reinstall of dependency chain on new branch.
  - Experiment with `cp -R` and `sed` failed.
- Hard to deploy branches to server.

Bumper

- After merging branch, need to bump versions.
- Bumping is transitive.
- `bumper`: tool for transitively bumping versions.
  - Uses `Cabal` library for finding and computing dependencies.
  - Regex for changing cabal files (preserves whitespace).

Bumper future

- Find libraries in need of bumping.
- Compute version bump according to PVP.
- Automatically deploy.

Forking

- Sometimes, need to fix 3rd party lib *now*!
- Haskell is open source: we can fix it.
- Public on our github (`http://github.com/silkapp`).
- Try to contribute back and remove.
- Deployed on private hackage, last version component ∼100.
- `darcs-fastconvert` for `darcs` repositories.

Other tools

- Continuous integration using `Jenkins`.
- Deployment using custom written tools.
- Custom web based tool for running processes.

Future plans

- Binary deployment (Linux and Mac?).
- Simpler solutions for everything.
- Open source more packages.

Contact

Interested?

- Check out Silk at `http://silkapp.com`.
- Email me at `erik@silkapp.com`.
- Follow us on twitter: @silkapp.
- Our github: `http://github.com`.

Contact

Interested?

- Check out Silk at `http://silkapp.com`.
- Email me at `erik@silkapp.com`.
- Follow us on twitter: @silkapp.
- Our github: `http://github.com`.

Questions?

Contact

Interested?

- Check out Silk at `http://silkapp.com`.
- Email me at `erik@silkapp.com`.
- Follow us on twitter: @silkapp.
- Our github: `http://github.com`.

Thanks!