Haskell Tools and Infrastructure

Matthias Fischmann, Alexander Ulrich

July 1, 2015 —

Topics

- How do I find a library for a given task?
- How do I install / manage those libraries?
- Where do I find answers to my questions?

Search engines

- `http://hayoo.fh-wedel.de/`
- `https://www.haskell.org/hoogle/`

Full-text search for packages, functions, types, ...

Many libraries are indexed.

You can set up your own instance:

- to reduce network load
- to index more / other packages (e.g. unreleased code)

http://hackage.haskell.org/

The official library distribution site.

- `https://github.com/haskell/hackage-server`
  the software to run hackage.haskell.org (or your own site)
- Think pypi (python), cpan (perl), ...

Features:

- cabal integration (next slide)
- web-based package search
- web-based documentation

http://www.stackage.org/

Package version constraint set that makes hackage more reliable.

- coherent nightly builds
- LTS

Cabal

- build environment

```
$ git clone ... && cd ... && cabal install
```

- library download and dependency manager

```
$ cabal install binding-gtk
$ cabal install haskell-in-space
$ cabal update
$ cabal list
```

Cabal

- integrates testing, docs generation, ...

```
$ cabal haddock
$ cabal test
```

- allows you to create your own packages

```
$ cabal check
$ cabal sdist
$ cabal upload
```

- just ask!

```
$ cabal --help
```

Think: make, ant, npm, bower, apt-get, ...

Cabal file

- Every package has a file:

  ```
  <package name>.cabal
  ```

  in the root directory.
- The cabal file provides info on
  - release
  - lincensing
  - maintainer
  - provided modules and executables
  - package dependencies
  - test suite
  - compiler settings

Cabal file

```
name:                gorbla
version:             8.109.11
-- synopsis:
-- description:
license:             AGPL-3
license-file:        LICENSE
author:              ...
maintainer:          ...
-- copyright:
-- category:
build-type:          Simple
cabal-version:       >= 1.10
```

Cabal file

```
library
  exposed-modules:
      Api
    , DB.Core
    [...]
    , Types
  build-depends:
      base >=4.7 && <4.8
    , acid-state >=0.12 && <0.13
    , aeson >=0.7 && <0.8
    [...]
    , warp >=3.0 && <3.1

executable thentos
  main-is:
      Main.hs
  build-depends:
      base
    , gorbla
```

ghc-pkg

To check what packages are installed:

```
$ ghc-pkg list
$ ghc-pkg list transformers
```

To remove local packages from database (not from disk!):

```
$ ghc-pkg unregister transformers
```

$\Rightarrow$ ”ghc-pkg unregister” beats ”cabal install –force-reinstall”

Cabal Sandboxes

```
$ git clone ... && cd ...
$ cabal sandbox init
$ cabal install
```

If you want to pull in packages from local source trees:

```
$ cabal sandbox add-source ~/.cabal-src/aeson/
```

To manage the packages installed in your sandbox:

```
$ cabal sandbox hc-pkg [list|unregister|...]
```

IDEs

`https://www.haskell.org/haskellwiki/IDEs`

Most commonly used:

- emacs
  - `https://github.com/haskell/haskell-mode/wiki`
    `https://www.haskell.org/haskellwiki/IDEs#Emacs`
  - Searchable speedbar, ctags integration
  - Incremental compilation and ghci integration
  - Lazy data exploration (`present`)
  - Code formatting, indentation, alignment
  - Code refactoring
- vi
  - `https://www.haskell.org/haskellwiki/IDEs#Vim`
- leksah
  - `http://leksah.org`

Other sources of information

If you want to aimlessly delve deeper into haskell:

- `http://learnyouahaskell.com/`
- `http://book.realworldhaskell.org/`
- `http://www.reddit.com/r/haskell/new`
- `https://ocharles.org.uk/blog`
- `http://haskellcast.com/`
- `http://well-typed.com/blog`

Other sources of information (cont.)

Places to look for answers to specific questions.

- `https://www.haskell.org/onlinereport/haskell2010/`
- `http://www.haskell.org/`
- `http://stackoverflow.com/questions/tagged/haskell`
- `irc:freenode#haskell-beginners,#haskell`

[Commercial break]

Well-Typed's Fast Track to Haskell

- 5th - 6th October 2015 in London, delivered by Andres Löh
- `https://skillsmatter.com/courses/464-well-typed-fast-track-to-haskell`