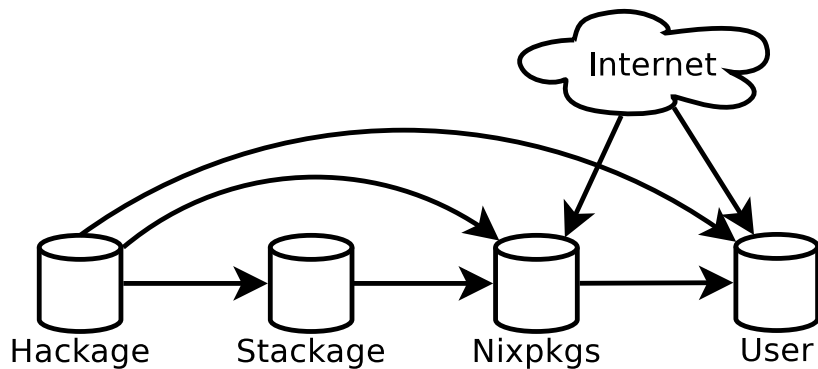


The Nixpkgs Haskell Infrastructure

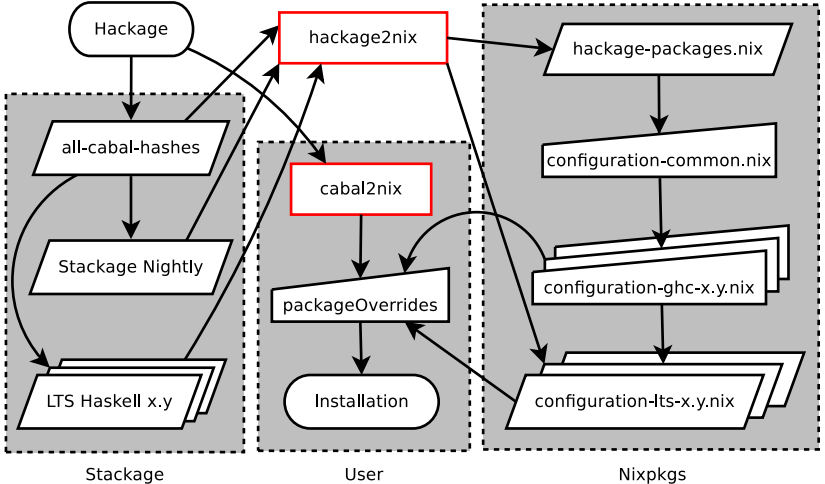
Peter Simons <simons@cryp.to>

NixCon 2015

Players Involved in Haskell Packaging

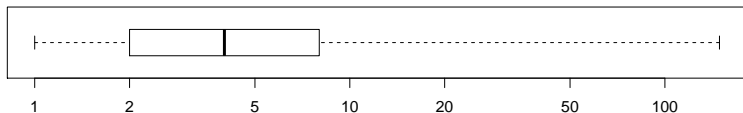


The Packaging Process



Hackage

- ▶ 62,423 Cabal files (+41.6 per day)
- ▶ 9,028 distinct packages (+4.9 per day)
- ▶ $\mu = 6.9, \sigma = \pm 10.3$ releases per package:



- ▶ Top 10 most active packages: egison (149), git-annex (145), hakyll (145), purescript (141), yesod-core (140), warp (136), lens (129), wai-extra (119), yesod (116), and hlint (114).

Hackage: Structure of the Git Repository

```
3d-graphics-examples
3dmodels
4Blocks
[...]
mtl
  +-- 1.0
  |   +-- mtl.cabal
  |   +-- mtl.json
  +-- [...]
  +-- 2.2.1
      +-- mtl.cabal
      +-- mtl.json
[...]
ztail
Zwaluw
```

Hackage: [mtl/2.2.1/mtl.cabal](http://hackage.haskell.org/package/mtl-2.2.1)

```
name:          mtl
version:       2.2.1
license:       BSD3
synopsis:      Monad classes, using functional dependencies
homepage:      http://github.com/ekmett/mtl
build-type:    Simple
```

Library

exposed-modules:

```
Control.Monad.Cont
[...]
```

build-depends: base < 6, transformers == 0.4.*

extensions:

```
MultiParamTypeClasses
FunctionalDependencies
FlexibleInstances
```

Hackage: mtl/2.2.1/mtl.json

```
{ "package-hashes" : { "MD5"           : "96a2f12b...",
                        "Skein512_512" : "73b5858d...",
                        "SHA1"          : "c244f8ec...",
                        "SHA256"        : "cae59d79...",
                        "SHA512"        : "5c31626b..."
                      },
  "package-locations" : [
    "https://hackage.haskell.org/package/mtl-2.2.1/" \
    "mtl-2.2.1.tar.gz",
    "https://s3.amazonaws.com/hackage.fpcomplete.com/" \
    "package/mtl-2.2.1.tar.gz"
  ],
  "package-size" : 15391
}
```

Hackage: Nix Build for mtl 2.2.1

```
{ mkDerivation, base, stdenv, transformers }:  
  
mkDerivation {  
  pname = "mtl";  
  version = "2.2.1";  
  sha256 = "cae59d79f3a16f8e9f3c9adc1010c7c6...";  
  libraryHaskellDepends = [ base transformers ];  
  homepage = "http://github.com/ekmett/mtl";  
  description = "Monad classes, using functional ...";  
  license = stdenv.lib.licenses.bsd3;  
}
```


Hackage: Destructive Editing

MonadRandom/0.4/MonadRandom.cabal contains a line:

```
x-revision: 2
```

This translates to Nix:

```
revision = "2";  
editedCabalFile = "2e218afd5b29c868...";
```

The build log will say:

*Replace Cabal file with edited version from
<http://hackage.haskell.org/package/MonadRandom-0.4/revision/2.cabal>.*

Hackage: Conditional Values

Values in Cabal files may depend on OS, architecture, compiler, and flags. For example, `hint 0.4.2.3` specifies:

Library

```
if impl(ghc >= 6.8)
  build-depends:    random, directory
  if impl(ghc >= 6.10)
    build-depends:  base >= 4, base < 5,
                   ghc-mtl == 1.2.1.*
  else
    build-depends:  base >= 3, base < 4
else
  build-depends:    utf8-string < 0.3
if !os(windows)
  build-depends:    unix >= 2.2.0.0
```

Hackage: Cabal Flags

Flags can be specified by the user or auto-configured by the build to fit the environment. For example, pandoc 1.15.1.1 defines:

Flag https

Description: Enable support for downloading over https.

Library

```
Build-Depends: base >= 4.2 && <5,  
               syb >= 0.1 && < 0.7,  
               [...]  
               ghc-prim >= 0.2
```

if flag(https)

```
Build-Depends: http-client >= 0.3.2 && < 0.5,  
               http-client-tls >= 0.2 && < 0.3,  
               http-types >= 0.8 && < 0.10
```

Stackage: Stable Hackage

- ▶ 1,605 packages (18% of Hackage)
- ▶ Build instructions published via Git specify parameters such as “don’t run test suite”, “don’t run Haddock”, etc.
- ▶ Testing takes place on 64-bit Linux.
- ▶ Nightly Snapshots contain the latest version of every package such that the package set is consistent.
- ▶ LTS minor releases are created weekly and contain only point releases that don’t change the API.
- ▶ LTS major releases contain API-breaking changes and usually coincide with a compiler update.

cabal2nix & hackage2nix

- ▶ Code lives at <https://github.com/NixOS/cabal2nix>:
 - ▶ `distribution-nixpkgs` library
 - ▶ `cabal2nix` executable
 - ▶ `hackage2nix` executable
- ▶ `update-nixpkgs.sh` script runs automatically once per hour.
- ▶ Changes go to `haskell-updates` branch in peti's Nixpkgs fork on Github.
- ▶ `hydra.cryp.to` continuously builds Stackage Nightly and the latest LTS Haskell package set.
- ▶ If Hydra builds work sufficiently well, updates are manually merged into `master` branch of main repository.
- ▶ `generate-nixpkgs-haskell-package-list` utility manually publishes new state of Nixpkgs on Hackage.

Nixpkgs: `hackage-packages.nix`

The package set is represented as a primitive recursive function:

```
ps = self: {  
    "3d-graphics-examples" = callPackage ...;  
    ...  
    "mtl_2_1_3_1" = callPackage ...;  
    "mtl" = callPackage ...;  
    ...  
    "ztail" = callPackage ...;  
};
```

The Nixpkgs attribute set `haskellPackages` refers to the fixpoint of that function which is computed by the expression:

```
let self = ps self;  
in      # = ps (ps self) = ps (ps (ps (self))) = ...  
    self
```

Nixpkgs: hackage-packages.nix

We can modify the package set using OO-style inheritance:

```
fix = f: let self = f self; in self;
extend = rattrs: f: self: let super = rattrs self;
                          in super // f self super;
```

```
ps = self: {
  foo = "foo"; bar = "bar";
  foobar = self.foo + self.bar;
};
```

```
f = self: super: { foo = reverse super.foo; };
```

- ▶ `(fix ps).foobar ≡ "foobar"`
- ▶ `(fix (extend ps f)).foobar ≡ "oofbar"`
- ▶ `(fix (extend (extend ps f) f)).foobar ≡ "foobar"`

Nixpkgs: pkgs/development/haskell-modules/default.nix

```
{ compilerConfig ? (self: super: {})  
  , packageSetConfig ? (self: super: {})  
  , overrides ? (self: super: {})  
  }:
```

```
fix  
  (extend  
    (extend  
      (extend  
        (extend haskellPackages commonConfiguration)  
        compilerConfig)  
      packageSetConfig)  
    overrides)
```


Nixpkgs: Multiple Views Of The Package Set

- ▶ `hackage-packages.nix` represents “Stackage Nightly”.
- ▶ `configuration-common.nix` “extends” that to work around deficiencies in `hackage2nix`.
- ▶ `configuration-ghc-x.y.nix` “extends” that with fixes relevant only for particular compiler versions:
`haskell.packages.ghcXY`.
- ▶ `configuration-lts-v.w.nix` “extends” those sets to choose default versions according to “LTS Haskell x.y”:
`haskell.packages.lts-x_y`.

Nixpkgs: overrideCabal

Haskell derivations take the `mkDerivation` function as an argument:

```
{ mkDerivation, ... }: mkDerivation {  
  pname = "mtl";  
  version = "2.2.1";  
  ...  
}
```

The `override` method changes the arguments passed to a derivation — so we can change the definition of `mkDerivation`:

```
overrideCabal = drv: f: drv.override (args: args // {  
  mkDerivation = drv: args.mkDerivation (drv // f drv);  
});
```

Nixpkgs: haskell.lib Helper Functions

```
doHaddock = ...;  
dontHaddock = drv: overrideCabal drv (drv: {  
  doHaddock = false;  
});
```

```
enableLibraryProfiling = ...;  
disableLibraryProfiling = drv: overrideCabal drv (drv: {  
  enableLibraryProfiling = false;  
});
```

```
addBuildTool = drv: x: addBuildTools drv [x];  
addBuildTools = drv: xs: overrideCabal drv (drv: {  
  buildTools = (drv.buildTools or []) ++ xs;  
});
```

Nixpkgs: Deep Overriding

```
fix = f: let x = f x // { __unfix__ = f; }; in x;

haskellPackages = self:
  let
    callPackage = callPackageWithScope self;

    callPackageWithScope = scope: drv: args:
      (stdenv.lib.callPackageWith scope drv args) // {
        overrideScope = f:
          callPackageWithScope
            (fix (extend scope.__unfix__ f))
            drv args;
      };
  in
  { "git-annex" = callPackage ...; };
```

Other Resources

- ▶ <http://hackage.haskell.org/>
- ▶ <http://www.stackage.org/>
- ▶ [http://nixos.org/nixpkgs/manual/
#users-guide-to-the-haskell-infrastructure](http://nixos.org/nixpkgs/manual/#users-guide-to-the-haskell-infrastructure)
- ▶ [http:
//lists.science.uu.nl/mailman/listinfo/nix-dev](http://lists.science.uu.nl/mailman/listinfo/nix-dev)
- ▶ #nixos at [#irc.freenode.org](irc://irc.freenode.org)