Conquer Cabal Hell with Nix

@steshaw

Overview of talk

1. Purely Functional Package Systems
2. Overview of Nix
3. How to use Nix with Haskell

Purely Functional Package Systems

Features

- multiple variants of a package at the same time (i.e. side by side)
- immutable packages
- atomic upgrades/rollbacks
- mutiple environments
- usable as non-root

Examples

- Nix
- Zero Install
- Listaller (was Autopackage)

Nix

- Source-based with "binary replacement".
- Scales up to entire Linux distribution. See NixOS.
- Written in C and Perl but looks like a Haskell port is in the works.
- Popular in the Haskell community.

Commands

Find packages

```
$ nix-env --query --available 'hello*'
$ nix-env -qa 'hello*'
```

Install a package

```
$ nix-env --install hello
$ nix-env -i hello
```

Remove a package