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

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

```
$ nix-env --uninstall hello
$ nix-env -e hello
```

COMMANDS (CONTINUED)

Alternatively rollback

```
$ nix-env --rollback
```

Upgrade your packages

```
$ nix-env --upgrade
```

ACCELERATING HASKELL DEVELOPMENT WITH NIX

demo time