

# Compiling to Nix

a quick overview

<https://functional.cafe/@yvan>

## Who?

- **PureNix** (Nix backend for the **PureScript** programming language)<sup>1</sup>
- `dhall-to-nix` (output of the **Dhall** configuration language)<sup>2</sup>
- Nix integration of the **Nickel** configuration language<sup>3</sup>
- **Guile** (while Guix relies on `nix-daemon`, AFAIK, there are no facilities to convert Guix packages to Nix packages)

---

<sup>1</sup><https://github.com/purenix-org/purenix>

<sup>2</sup><https://github.com/dhall-lang/dhall-haskell>

<sup>3</sup><https://github.com/nickel-lang/organist>

# *Why? Types!*

*Why might someone choose not to write an expression directly in Nix?*

Writing correct and maintainable large Nix expressions is hard.  
Once again, types to the rescue!

E.g. There is the PoC of a cabal2nix<sup>4</sup> clone that does not rely on IFD<sup>5</sup> written in PureScript.

---

<sup>4</sup><https://github.com/cdepillabout/cabal2nixWithoutIFD>

<sup>5</sup>Import From Derivation

## *What for?*

Can I use language  $X$  to write:

- a Nix package?
- a developer shell?
- NixOS options?
- an entire NixOS configuration?
- a Flake?

## Checklist

- Can I embed a Nix expression within it?
- Can I embed it within a Nix expression?

## Checklist: Dhall

- Can I embed a Nix expression within it?
- Can I embed it within a Nix expression?

Dhall cannot encode many common Nixpkgs/NixOS idioms, such as:

- general recursion (e.g., `pkgs.callPackages`, the overlay system, and the NixOS module system)
- weakly-typed conversions (like `builtins.listToAttrs`)
- row polymorphism (i.e., the ... in { foo, bar, ... })

## Checklist: PureScript

- ☒ Can I embed a Nix expression within it?
- ☒ Can I embed it within a Nix expression?

PureScript allows you to import an arbitrary Nix expression as “*FFI*”. **But**, you need to ensure everything aligns with the strict type system.

PureScript offers its own ecosystem of packages (managed by spago) that can be harnessed to produce readable and well-organized Nix files. **But**, the generated code carries significant abstraction overhead and will always be an attribute set of what the module exposes.

## Checklist: Nickel

- Can I embed a Nix expression within it?
- Can I embed it within a Nix expression?

There is an **experimental** Nix toolkit to use nickel as a language for writing nix packages, shells and more.

**But**, AFAIU, Nickel does not compile to Nix!

## Open questions

What languages do we commit in `nixpkgs`? Nix? Bash? Nickel?  
etc.

Q/A