4 Branches **0** Tags

Go to file

Go to file

elpdt852 Add incremental demo.gif and examples/incre… ✓

a3ed6ed · yesterday

.github/workflows	Add basic CI (#22)	yesterday
crates	Add incremental dem…	yesterday
docs	Add incremental dem…	yesterday
examples	Add incremental dem…	yesterday
.envrc	Initial commit for open…	5 days ago
.gitignore	Initial commit for open…	5 days ago
CONTRIBUTI…	Add developing locall…	3 days ago
Cargo.lock	refactor: Move to mini…	yesterday
Cargo.toml	Initial commit for open…	5 days ago
LICENSE	Initial commit for open…	5 days ago
README.md	Add incremental dem…	yesterday
default.nix	Initial commit for open…	5 days ago
flake.lock	Initial commit for open…	5 days ago
flake.nix	Add incremental dem…	yesterday
mkMesonPack…	Initial commit for open…	5 days ago
shell.nix	Initial commit for open…	5 days ago

About

Ninja-compatible incremental build system for Nix

#nix #nixos #ninja #build-system

- Readme
- MIT license
- Activity
- Custom properties
- **92** stars
- **7** watching
- **2** forks

Report repository

Releases

No releases published

Packages

No packages published

Contributors 3

- **elpdt852** Edgar Lee
- **mupdt** Matej Urbas
- **gbpdt** Graham Bennett

Languages

README MIT license

nix-ninja

Incremental compilation of Ninja build files using Nix Dynamic Derivations.

Choosing ninja as the build graph representation lets us support any build system that outputs ninja like CMake, meson, premake, gn, etc.

[Key features](#key-features) • [Getting started](#getting-started) • [Design notes](#design-notes) • [Contributing](#contributing)

Key features

> Important
>
> There are still major todos, and depends on experimental features from an unreleased version of Nix. Come help us get nix-ninja to be useful day-to-day and working with an official Nix release!

- Parses `ninja.build` files and generates a derivation per compilation unit.
- Stores build inputs & outputs in content-addressed derivations for granular and Nix-native incrementality.
- Compatible CLI for ninja, so if you set `$NINJA` to `nix-ninja` then meson just works.
- Supports running locally (which runs `nix build` on your behalf), or inside a Nix derivation (which creates dynamic derivations).

Getting started

First you need to use [nix@d904921](#) and enable the following experimental features:

```
experimental-features = ["nix-command" "dynamic-derivations" "ca-derivations" "recursive-nix"]
```

Then you can try building the examples:

```
# Builds a basic main.cpp.
nix build github:pdtpartners/nix-ninja#example-hello

# Builds a basic main.cpp with dependency inference for its header.
nix build github:pdtpartners/nix-ninja#example-header

# Builds Nix 2.27.1.
nix build github:pdtpartners/nix-ninja#example-nix
```

You can also try running `nix-ninja` outside of Nix, but you'll need both `nix-ninja` and `nix-ninja-task` to be in your `$PATH`. Make sure `nix-ninja-task` is from the `/nix/store` as it is needed inside derivations `nix-ninja` generates.

```
export NIX_NINJA=$(nix build --print-out-paths)
export PATH="${NIX_NINJA}/bin:$PATH"
# Meson respects this environment variable and uses it as if its ninja.
export NINJA="${NIX_NINJA}/bin/nix-ninja"
```

Then you can go to an example and run it like so:

```
$ nix-shell
$ cd examples/hello
$ meson setup build
$ cd build
$ meson compile hello
$ ./hello
Hello Nix dynamic derivations!
```

Contributing

We still have major TODOs, so would appreciate any help. We've organize them under two GitHub milestones:

- `0.1.0` - The first release of `nix-ninja` aiming for correctness.
- `0.2.0` - Major performance features to make incremental builds productive.

Regardless, pull requests are welcome for any changes. Consider opening an issue to discuss larger changes first, especially when the design space is large.

Please read CONTRIBUTING and the design notes so you understand the big picture and prior art.

License

The source code developed for nix-ninja is licensed under MIT License.