



Nix, but with a different frontend

Théophane Hufschmitt — NixCon 2023



nickel-lang / **organist**

[Code](#) [Issues 11](#) [Pull requests 3](#) [Actions](#) [Projects 1](#) [Wiki](#) [Security](#) [Insights](#) [...](#)

organist Public [Edit Pins](#) [Unwatch 13](#) [Fork 10](#) [Star 186](#)

[main](#) [Go to file](#) [Add file](#) [Code](#) **About**

[Branches](#) [Tags](#)

thufschmitt Merge pull request [#114](#) from nickel-lang/run-test... [...](#) 3 hours ago 288

.github	Trigger the CI on merge queues	last month
doc-ideal	Document the Nix integration in slightly more detail	2 months ago
examples/c-hello-world	Merge the different kind of Nixel _things_	last month
future	Format all Nickel files with Topiary	4 months ago
lib	Separate the builders and the shells	last month
notes	Save drafts of brainstorming session on builders	8 months ago
templates/default	Merge the different kind of Nixel _things_	last month
.gitignore	Add a shell example using nickel-nix	last year

About

An experimental Nix toolkit to use nickel as a language for writing nix packages, shells and more.

- [Readme](#)
- [MIT license](#)
- [Activity](#)
- 186 stars**
- 13 watching**
- 10 forks**

[Report repository](#)

Releases

No releases published

[Create a new release](#)







MODULES ARE *TRULY* GREAT

And not taken seriously enough

Built-in merging = (and)



Built-in merging = (and)



```
with import <nixpkgs> {};  
  
(lib.evalModules {  
  modules = [  
    ({ config, ... }:  
      {  
        options = {  
          foo = lib.mkOption { type = lib.types.int; };  
          bar = lib.mkOption { type = lib.types.int; };  
        };  
        config = {  
          foo = 2;  
          bar = config.foo;  
        };  
      })  
    ];  
  {  
    config.foo = lib.mkForce 1;  
  }  
}).config
```

Built-in merging = (and)



```
with import <nixpkgs> {};  
  
(lib.evalModules {  
  modules = [  
    ({ config, ... }:  
      {  
        options = {  
          foo = lib.mkOption { type = lib.types.int; };  
          bar = lib.mkOption { type = lib.types.int; };  
        };  
        config = {  
          foo = 2;  
          bar = config.foo;  
        };  
      })  
    ];  
  config.foo = lib.mkForce 1;  
}).config
```

```
{
```

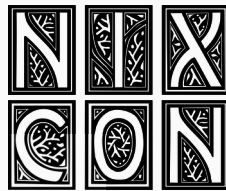
```
foo | Number = 2,  
bar | Number = foo,
```

```
&
```

```
{
```

```
foo | force = 1
```

```
}
```

```
let inputs = import "./nickel.lock.ncl" in
let organist = inputs.organist in

{
  shells = organist.shells.Bash,

  shells.build = {
    packages = {},
  },

  shells.dev = {
    packages.hello = organist.lib.import_nix "nixpkgs#hello",
  },
} | organist.contracts.NixelExpression
```



So what?

One abstraction to leak them all



```
{  
  name = "hello",  
  version = "1.2.3"  
  src = fetch_url & {  
    url = "https://hello.com/%{name}-%{version}.tar.xz",  
    hash = "sha256-Pld6Ro2GMPxFh3+oVfFtj/YCnoUX0Q9uET2LS2fx2gM=",  
  },  
}
```

One abstraction to leak them all



```
{
  name = "hello",
  version = "1.2.3"
  src = fetch_url & {
    url = "https://hello.com/{name}-{version}.tar.xz",
    hash = "sha256-Pld6Ro2GMPxFh3+oVfFtj/YCnoUX0Q9uET2LS2fx2gM=",
  },
}

&
{
  version = "1.2.4",
  src.hash = "sha256-dj+TzhwuPmebcZryVAhmmvZCzYLKAC7Jln6Zi0lmzX8=",
}
```



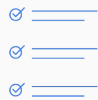
So what?



Subscribe to
Tweak!

TWEAG
by Modus Create

Tweak's Nix Services:



Assessment



Initial set up
and
migration



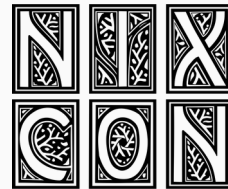
Training



Tuning &
maintenance



Upstreaming

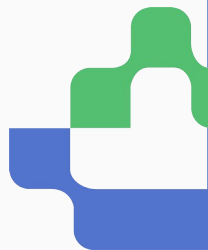
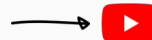


Come meet our team! [Tweak is a Gold sponsor.](#)

Learn more about Tweak and our extended capabilities
now that we're a part of the Modus Create platform.



You can also check out **The Nix Hour** on
YouTube.

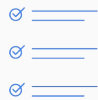




Subscribe to
Tweag!

TWEAG
by Modus Create

Tweag's Nix Services:



Assessment



Initial



Tuning &
maintenance



Upstreaming

Join our team! [Tweag is a Gold sponsor.](https://github.com/nickel-lang/organist)

Learn more about Tweag and our extended capabilities
now that we're a part of the Modus Create platform.



You can also check out **The Nix Hour** on
YouTube.

