

Calling Nix from Python

Calling Python from Nix

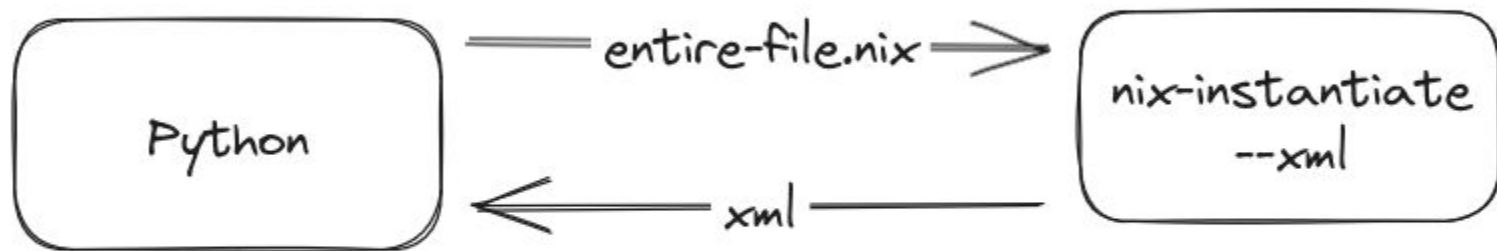
Yorick van Pelt - Tweag IO

<https://pub.yori.cc/nixcon23.pdf>

Calling Nix from Python



NixOps et al.

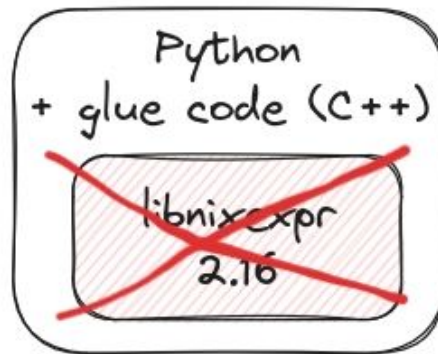
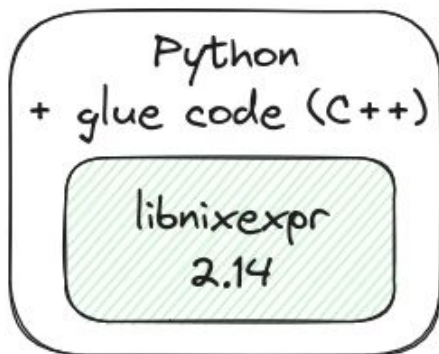


- stable API
- slow: unnecessary evals or duplicate work

Calling Nix from Python



Pythonix

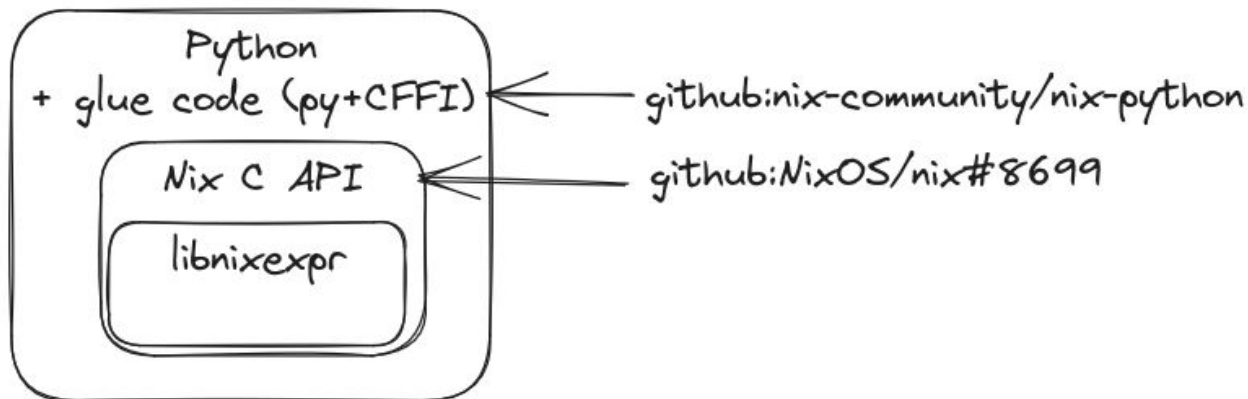


- unstable API
- maintainer burden (specifically for python)
- fast

Calling Nix from Python



The idea



- Stable API
- less maintainer burden
(universal, incremental)
- still fast
- solves this forever

Why a C API?



- C isn't a programming language anymore
- Easy interfaces to Rust, Swift, Haskell, Python, Javascript

Making the C API



- Developed in tandem with the Python API
- Oh no

✓ CI / tests (ubuntu-latest) (pull_request) Successful in 41m

edolstra commented on Apr 24

Member



This fails on macOS with:

```
> libc++abi: terminating with uncaught exception of type nix::SysError: error: i
```



roberth changed the title ~~Stable C bindings for libutil, libexpr~~ (Towards) stable C bindings for libutil, libexpr on Jul 18

[tweag/nix] Run failed: CI - nix-c-bindings (919ee3d) - CI workflow... Aug 28
[tweag/nix] Run failed: CI - nix-c-bindings (cb1c308) - CI workflow... Aug 28

[tweag/nix] Run failed: CI - nix-c-bindings (029deab) - CI workflow... Jul 30
[tweag/nix] Run failed: CI - nix-c-bindings (869d7d6) - CI workflow... Jul 30
[tweag/nix] Run failed: CI - python-bindings-cffi (9985b60) - CI ... Jul 14
[tweag/nix] Run failed: CI - python-bindings-cffi (a6228e2) - CI ... Jul 5
[tweag/nix] Run failed: CI - python-bindings-cffi (596f8b4) - CI ... Jul 5

[tweag/nix] Run failed: CI - python-bindings-cffi (72f669e) - CI ... Jun 29
[tweag/nix] Run failed: CI - python-bindings-cffi (f4ff80a) - CI w... Jun 28
[tweag/nix] Run failed: CI - python-bindings-cffi (37af65a) - CI ... Jun 21
[tweag/nix] Run failed: CI - python-bindings-cffi (994ee71) - CI ... Jun 21
[tweag/nix] Run failed: CI - python-bindings-cffi (c44a4ac) - CI ... Jun 21
[tweag/nix] Run failed: CI - python-bindings-cffi (1795935) - CI ... Jun 21
[tweag/nix] Run failed: CI - python-bindings-cffi (1babe43) - CI ... Jun 19
(6d63a36) - CI ... Jun 19
(2534b56) - CI ... Jun 19
+2,883 -12 (0e21fb1) - CI ... Jun 19
(4a6b0bd) - CI ... Jun 18

[tweag/nix] Run failed: CI - python-bindings-cffi (4de2114) - CI ... Jun 2
[tweag/nix] Run failed: CI - python-bindings-cffi (3389f95) - CI ... May 26
[tweag/nix] Run failed: CI - python-bindings-cffi (aa8ce9b) - CI workflow... May 26
[tweag/nix] Run failed: CI - macos-sandbox (a465616) - CI workflow... May 11
[tweag/nix] Run failed: CI - macos-sandbox (c6dc98b) - CI workflow... May 8
[tweag/nix] Run failed: CI - python-bindings-cffi (f6a0320) - CI ... May 8
[tweag/nix] Run failed: CI - python-bindings-cffi (604b9ad) - CI ... May 8
[tweag/nix] Run failed: CI - python-bindings-cffi (eae962f) - CI ... May 8
[tweag/nix] Run failed: CI - python-bindings-cffi (ffe0fd9) - CI workflow... Apr 29
[tweag/nix] Run failed: CI - python-bindings-cffi (c0e9b2e) - CI ... Apr 28
[tweag/nix] Run failed: CI - python-bindings-cffi (7a320e0) - CI ... Apr 25
[tweag/nix] Run failed: CI - python-bindings-cffi (25b978f) - CI ... Apr 25
[tweag/nix] Run failed: CI - python-bindings-cffi (7658eb6) - CI ... Apr 24
[tweag/nix] Run failed: CI - python-bindings-cffi (4aa524d) - CI ... Apr 24

Making the C API



- Developed in tandem with the Python API
- Error handling
- Making two garbage collectors work together

What can it do



- Parsing and evaluating Nix code
 - `nix_expr_eval_from_string(evaluator, "builtins.nixVersion", value);`
- Inspecting values and calling functions
 - `printf("nix version: %s\n", nix_get_string(value));`
- Modifying and creating values and functions
 - `void increment(void*, State*, Value* args[], Value* ret) {
 nix_set_int(ret, nix_get_int(args[0]) + 1);
}`
`nix_register_primop(nix_alloc_primop(increment, 1, "increment", ...))`

C API usage



```
nix_expr_eval_from_string(nil, evaluator, "builtins.nixVersion", value);
```

```
return 0;
```

Error Handling



```
nix_c_context* ctx = nix_c_context_create();  
  
nix_expr_eval_from_string(ctx, evaluator, "builtins.nixVersion", value);  
  
if (nix_err_code(ctx) != NIX_OK) goto error;  
  
return 0;  
  
error:  
  
printf("error: %s\n", nix_err_msg(NULL, ctx, NULL));
```

Garbage Collection



```
nix_c_context* ctx = nix_c_context_create();  
  
nix_expr_eval_from_string(ctx, evaluator, "builtins.nixVersion", value);  
  
if (nix_err_code(ctx) != NIX_OK) goto error;  
  
nix_gc_decref(NULL, value);  
  
return 0;  
  
error:  
  
printf("error: %s\n", nix_err_msg(NULL, ctx, NULL));  
  
if (value) nix_gc_decref(NULL, value);
```

The Python API



- CFFI: Reads header files, generates C code
- Python: Wrapper classes for Nix Values, functions, Store

```
import nix
pkgs = nix.eval("import <nixpkgs> {}")

print(len(pkgs))
print(pkgs["hello"].build())

hello2 = pkgs["hello"]["overrideAttrs"](lambda o: {
    "pname": str(o["pname"]) + "-test"
})
print(hello2.build())
```

PYTHON



Calling Python from Nix

Nix plugins



```
nix eval \  
--plugin-files ./my-plugin.so
```

my-plugin.so

Nix plugins



- Documented!
- [plugin-files](#)
 - A list of plugin files to be loaded by Nix. Each of these files will be dlopened by Nix, allowing them to affect execution through static initialization. In particular, these plugins may construct static instances of RegisterPrimOp to **add new primops or constants to the expression language**, RegisterStoreImplementation to **add new store implementations**, RegisterCommand to **add new subcommands** to the `nix` command, and RegisterSetting to **add new nix config settings**. See the constructors for those types for more details.
 - Warning! These APIs are inherently unstable and may change from release to release.

Nix plugins



- Documented!
- [plugin-files](#)
 - A list of plugin files to be loaded by Nix. Each of these files will be dropped by Nix, allowing them to affect execution through static initialization. In particular, these plugins may construct static instances of RegisterPrimOp to **add new primops or constants to the expression language**, RegisterStoreImplementation to **add new store implementations**, RegisterCommand to **add new subcommands** to the nix command, and RegisterSetting to **add new nix config settings**. See the constructors for those types for more details.
 - ~~Warning! These APIs are inherently unstable and may change from release to release.~~

Nix plugins - Example



- Real example: haskell.nix + SimdJSON

```
$ time nix-2.11 eval --json -f testcase2.nix yellor  
{..  
Executed in 613.84 millis
```

```
$ time nix --plugin-files simdjson.so eval --json -f testcase2.nix yellor  
{..  
Executed in 124.87 millis
```

Python plugin



```
$ nix repl --plugin-files ./pyImport.so  
> c = builtins.pyImport "b64"  
> c.encode "hello"  
"aGVsbG8="
```



```
def hello(pkgs):  
    return pkgs["stdenv"]["mkDerivation"]({  
        "name": "hello",  
        "src": pkgs["hello"]["src"]  
    })
```

Try it at home



```
$ nix develop github:tweag/python-nix#env
```

```
$ python
```

```
>>> import nix
```

```
>>> nix.eval(...)
```



Hackathon ideas

- Stability: use and feedback!
- Port or make CI and build tools
- Store API
- Javascript, Rust + others
- WASM imports



A Modus Create Company

Software
Innovation Lab



antithesis

<https://pub.yori.cc/nixcon23.pdf>





Subscribe to
Tweag!

TWEAG
by Modus Create

Tweag's Nix Services:



Assessment



Initial set up
and migration



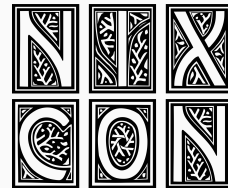
Training



Tuning &
maintenance



Upstreaming

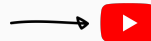


Come meet our team! [Tweag is a Gold sponsor.](#)

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.



These slides: <https://pub.yori.cc/nixcon23.pdf>