Updating Tools

To update tools bundled with OpenLane, their respective `.nix` files need to be updated. The relevant `.nix` files all exist under the `nix/` folder under the root of the OpenLane repo.

Setting up the OpenLane Nix Cache

If you somehow haven't done so already, follow the instructions in **Nix Installation** to enable the OpenLane-specific binary cache.

Finding the Current Version Info

First, find the `.nix` file for your tool. For OpenROAD, for example, it is `openroad.nix`.

There are four primary values in the nix object (called a "derivation") that decide the version being used (all strings):

- `owner`: The name of the GitHub tool repository owner.
- `repo`: The name of the GitHub tool repository.
- `rev`: By convention, the Git commit of the tool in question.
  - Technically, versions, tags and branches are also accepted, but none are deemed consistent enough to use with OpenLane.
- `sha256` (or `hash`): A sha256sum of the contents of the repository in use when said commit is downloaded without any `.git` files.

Let's take `openroad.nix` as an example:

```
    inherit rev;

    src = fetchFromGitHub {
      owner = "The-OpenROAD-Project";
      repo = "OpenROAD";
      inherit rev;
      inherit sha256;
    };
```

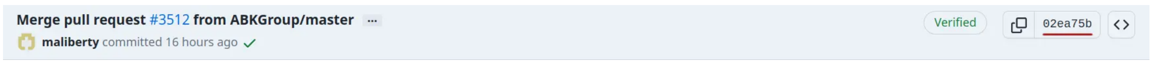
latest

Changing Version Info

Start by doing two things:

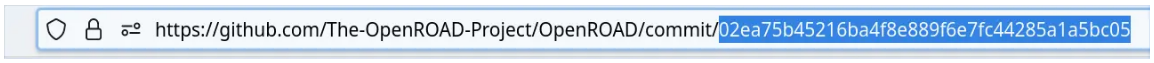
1. Finding the version you want to update to in the target repository.

Since we're updating OpenROAD, going to https://github.com/The-OpenROAD-Project/OpenROAD/commits/master will give us a list of commits. The easiest way to get the commit hash in question is to just click on a commit…



Merge pull request #3512 from ABKGroup/master ···
maliberty committed 16 hours ago ✓
Verified 02ea75b

…and then copy the hash from the URL:



https://github.com/The-OpenROAD-Project/OpenROAD/commit/02ea75b45216ba4f8e889f6e7fc44285a1a5bc05

2. Updating the version in the Nix derivation

Change the value of `rev` to match the hash you just copied, and then set the value of `sha256` to an empty string (`""`).

Warning

Do not leave the `sha256` field unchanged. Nix attempts to find downloaded files in its cache by hash first, and if the hash does not change it will not attempt to re-download the Git repository, essentially ensuring you will waste time re-building a Git repository you've previously downloaded, while substituting a mismatched revision string to boot.

3. Invoke `nix-shell` and wait. Depending on the tools's repository size and your internet connection, this will take a while.

You will then encounter an error that looks something like this:

```
error: hash mismatch in fixed-output derivation '/nix/store/2x8jfn70v4ahvvcdbc
         specified: sha256-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
            got:    sha256-zQVs4u0x1Lh1gs0xm9eJHc36LQNzlWFMpLVRzOOdhmU=
```

Copy the value from "got": in this case, `sha256-zQVs4u0x1Lh1gs0xm9eJHc36LQNzlWFMpLVRzOOdhmU=`, and use it as the value for `sha256`:

```
sha256 = "sha256-zQVs4u0x1Lh1gs0xm9eJHc36LQNzlWFMpLVRzOOdhmU=";
```

4. Invoke `nix-shell` again

You'll notice that the Git clone was not repeated, and the build should start.

That's pretty much it. Just wait and the resulting nix shell will contain the new version of the utility.

If it errors out, this may be because of one of the following:

- The build methodology for the tool has changed
  - You will need to update more elements in the Nix derivation, which will require understanding Nix. See **Further Reading** below for more information on how to learn to do that.
- An internet connectivity issue
  - Nix will automatically download derivations it needs from various servers, including but not limited to the Nix Store, Cachix, and GitHub. If there is a temporarily failure accessing any of them, the build will fail.
- Memory corruption or other hardware failure
  - Restarting the build should work, build you can also try `nix-shell --cores 1` to limit the stress on your hardware.

Further Reading

See Nix under **Contributing Code** for more information about our conventions for Nix derivations.

To gain a deeper understanding of the Nix programming language, try Nix Pills.

latest