Installing Every NixOS Package

Anthony Wang (xy)

2026-03-07

Why?

Why?

Uh…

Why?

Uh…

Why not?

Why?

Uh…

Why not?

Previously, I installed every Arch package.

Attempt 1

Easy! Just:

```
environment.systemPackages = builtins.attrValues pkgs;
```

Attempt 1

Easy! Just:

```
environment.systemPackages = builtins.attrValues pkgs;
```

Nope! `pkgs` has tons of broken packages and junk.

Attempt 2

Solution: @untrusem suggested using Hydra to identify broken packages

Attempt 2

Solution: @untrusem suggested using Hydra to identify broken packages

But we're not done yet! Need to include second-level packages like `pkgs.kdePackages.konsole`.

Attempt 3

Get list of all packages:

```
let
  pkgs = import <nixpkgs> { };
  lib = pkgs.lib;
  getpkgs =
    y: a: b:
    builtins.map (
      x:
      if
        (builtins.tryEval x.value).success
        && builtins.isAttrs x.value
        && !lib.strings.hasPrefix "pkgs" x.name # Ignore pkgs.pkgs*
        && !lib.strings.hasPrefix "__" x.name # Bad stuff
        && x.name != y.name # Definitely infinite recursion
        && x.name != "lib" # Ignore pkgs.lib, pkgs.agdaPackages.lib, and other evil stuff
        && x.name != "override" # Doesn't contain packages
        && x.name != "buildPackages" # Another copy of pkgs
        && x.name != "targetPackages" # Yet another copy of pkgs
        && x.name != "formats" # Ignore the pkgs.formats library
        && x.name != "tests" # Ignore tests
        && x.name != "nixosTests" # Ignore more tests
        && x.name != "scope" # Ignore
pkgs.haskell.packages.ghc910.buildHaskellPackages.generateOptparseApplicativeCompletions.scope
which contains another copy of pkgs
        && x.name != "_cuda" # Proprietary garbage
        && x.name != "vmTools" # Not a VM
        && x.name != "ghc902Binary" # Broken
        && x.name != "coqPackages_8_11" # Broken
        && x.name != "coqPackages_8_12" # Broken
        && x.name != "pypyPackages" # Broken
        && x.name != "pypy2Packages" # Broken
        && x.name != "pypy27Packages" # Broken
      then
        (
          if x.value ? "drvPath" then
            (
              if (builtins.tryEval x.value.drvPath).success then
                b
                + "|"
                + x.name
                + " "
                + (if x.value ? "pname" then x.value.pname else "unnamed")
                + " "
                + x.value.outPath
              else
                [ ]
            )
          else if a > 10 then
            abort "Probably infinite loop?"
          else
            builtins.trace a
            <| builtins.trace x.name
            <| builtins.trace b
            <| getpkgs x (a + 1)
            # For some stupid reason x.name can contain . so use | as the separator instead
            <| b + "|" + x.name
        )
      else
        [ ]
    )
    <| lib.attrsToList y.value;
in
lib.strings.concatStringsSep "\n"
<| lib.lists.flatten
<| getpkgs {
  name = "pkgs";
  value = pkgs;
} 0 "pkgs"
```

Next steps

2. Filter out duplicates

Next steps

2. Filter out duplicates
3. Filter broken packages by querying cache.nixos.org (the Hydra trick doesn't scale very well)

Next steps

2. Filter out duplicates
3. Filter broken packages by querying cache.nixos.org (the Hydra trick doesn't scale very well)
4. Override a module to fix `pkgs.buildEnv` errors (thanks @ersei for the help)

Next steps

2. Filter out duplicates
3. Filter broken packages by querying cache.nixos.org (the Hydra trick doesn't scale very well)
4. Override a module to fix `pkgs.buildEnv` errors (thanks @ersei for the help)
5. Install them all!!!

Oops!


```
error: your NixOS configuration path seems to be missing
essential files.
To avoid corrupting your current NixOS installation, the
activation will abort.

If you think this is a mistake, you can set the environment
variable
NIXOS_REBUILD_I_UNDERSTAND_THE_CONSEQUENCES_PLEASE_BREAK_MY_SYSTEM
to 1
and re-run the command to continue.
Please open an issue if this is the case.
```

Stats

- 71440 packages
- 1.34 TB `/nix/store`
- Only 690 GB on disk thanks to compression

Demo!

Questions?

Check out this blog post for more info