

home-manager

Use [Nix](#) to manage your user environment.

<https://github.com/nix-community/home-manager> 

Getting Started

<https://github.com/juspay/nix-dev-home> 

To view help from terminal,

```
$ man home-configuration.nix
```

Sub-pages

Folgezettel children

[home-manager Tutorial Series](#)

[Separate Git “profiles”](#)

[Installing fonts using home-manager](#)





Links to this page

[home-manager Tutorial Series](#)

A tutorial series on home-manager,

[direnv: manage dev environments](#)

If you use home-manager, both `nix-direnv` and `starship` can be installed using the following configuration:

[!tip] Newcomer Tip If you have never used home-manager before, we recommend that you set it up by following the instructions at <https://github.com/juspay/nixos-unified-template>  (which is based on [nixos-unified](#) , thus works on macOS and Linux).

[Using home-manager to manage dotfiles, packages, services](#)

Sridhar Ratnakumar[↗] will demonstrate using Nix to declaratively manage your **dotfiles, programs and services** using home-manager, thus replacing the likes of legacy software like homebrew. The goal is to create an **one-click environment** to setup an user environment on any system, be it a Macbook or a Linux machine.

Separate Git “profiles”

When using home-manager, you can add the following to your `programs.git` module:

Installing fonts using home-manager

Whethere you are on macOS or NixOS, you can install and setup fonts in an unified fashion with Nix using home-manager.

Install Nix

home-manager (and nix-darwin if you are on macOS) if you would like to use Nix for more than packages and devShells.

Git

Git can be declaratively configured in Nix via home-manager. Here is[↗] an example:

