

# Contents

<b>Prefatory Matters</b>	<b>1</b>
<b>Dependencies</b>	<b>1</b>
<b>Components</b>	<b>1</b>
dwl . . . . .	1
somebar . . . . .	2
Installation . . . . .	2
Configuration . . . . .	2
someblocks . . . . .	2
cage . . . . .	2
Installation . . . . .	2
Usage . . . . .	2
swhkd . . . . .	2
Installation . . . . .	3
Usage . . . . .	3
<b>Post-Installation</b>	<b>3</b>

## Prefatory Matters

DWL is a fork of DWM that runs on Wayland. Refer to the github pages of all the applications included in order to install the programs and their respective dependencies properly. Nonetheless, I'll provide most of the basic commands required to setup my dotfiles. Once my distro-hopping days are over, I'll upload a shell-script that automates the task of setting up my dotfiles. Maybe, in a month or so, I'll upload a shell-script for VoidLinux users. With my dotfiles, I have also included a fork of rofi that supports Wayland as well as Cage, a Wayland based kiosk compositor, that I use to run programs with my Nvidia card.

## Dependencies

wayland dunst kitty acpi pipewire pipewire-pulse wlroots gcc base-devel flatpak (optional) nvidia

## Components

### dwl

To compile `dwl`, execute the following commands:

```
$ cd dwl
$ make
```

```
# make clean install
```

**Disclaimer:** Dwl usually comes with Xwayland support disabled. If you do not wish to use Xwayland, edit the config.mk file accordingly (The config.mk file is pretty self-explanatory).

## somebar

### Installation

```
$ cd somebar
$ cp src/config.def.hpp src/config.hpp
$ meson setup build
$ ninja -C build
# ninja -C build install
```

### Configuration

```
$ vim src/config.hpp; ninja -C build ; sudo ninja -C build install
```

## someblocks

### Installation

```
$ cd someblocks
$ make
# make clean install
```

## cage

### Installation

To build cage, the meson build system is required. Apart from that it requires wayland, wlroots, and xkbcommon to be on the system. Build the kiosk by following these steps:

```
$ cd cage
$ meson build -Dxwayland=true --buildtype=release
$ ninja -C build
```

### Usage

```
$ pathtocage/build/cage [program]
```

## swhkd

swhkd is a display protocol-independent hotkey daemon made in Rust. swhkd uses an easy-to-use configuration system inspired by sxhkd so you can easily add or remove hotkeys.

## Installation

```
make setup  
make clean  
sudo make install
```

## Usage

```
swhks &  
pkexec swkbd
```

swkbdrc must be placed in /etc/swkbd

## Post-Installation

Execute the script startwayland