

# rdu's dotfiles handbook

2026-06-05

## Contents

Setup guide .....	2
Fresh machine .....	3
The stow model .....	3
Services .....	3
Tool versions: pacman + mise .....	3
Day to day .....	3
Docs .....	4
Justfile reference .....	4
Top-level .....	4
Install: essentials .....	5
Install: graphical stack .....	5
Install: optional extras .....	5
Stow (symlink management) .....	5
Services .....	5
niri: reload helpers .....	6
Maintenance .....	6
Docs (this site) .....	7
tmux shortcuts .....	7
Custom bindings .....	7
After the prefix .....	7
No prefix (root table) .....	8
Copy mode (vi keys) .....	8
OFF mode .....	9
tmux defaults worth remembering .....	9
Plugin bindings (TPM) .....	9
copycat .....	10
yank .....	10
extrakto (fuzzy search) .....	10
resurrect + continuum .....	10
pain-control (panes) .....	10
sessionist (sessions) .....	11
sessionx (session picker) .....	11
open .....	11
thumbs .....	11
logging .....	12
fzf .....	12
zellij shortcuts .....	12
Tmux mode – Ctrl a .....	13
Always available (normal & locked) .....	14

Pane mode — p .....	14
Tab mode — t .....	15
Resize mode — r .....	15
Move mode — m .....	16
Scroll mode — s .....	16
Scrolling keys (scroll & search) .....	16
Search options .....	17
Session mode — o .....	17
Locked mode (default) .....	17
Mode switches & misc .....	17
niri shortcuts .....	18
Apps & launchers .....	18
Mouse warp (wl-kbptr) .....	19
Media, volume & brightness .....	20
Focus .....	20
Moving windows & columns .....	21
Workspaces .....	22
Monitors .....	24
Layout & sizing .....	25
Screenshots & screencasting .....	25
Session & misc .....	26
Shell shortcuts (fish) .....	26
Custom bindings .....	26
thefuck — fix the last command .....	26
History — atuin .....	26
fzf.fish pickers .....	27
fish built-ins worth remembering .....	27
Handy abbreviations & functions .....	27
wl-kbptr .....	28
Install .....	28
Binds .....	28
Mid-stream click keys .....	29
Modes .....	29
Cell-pick keys (home_row_keys) .....	29
Universal keys .....	30
Useful CLI flags .....	30
Multi-monitor notes .....	30
Gotchas .....	30
Files .....	30

## Setup guide

How a fresh CachyOS/Arch machine becomes one of mine. Everything is driven by just — run just from the repo root to list every recipe (full list in the justfile reference).

## Fresh machine

```
git clone git@github.com:rdu/dotfiles.git ~/.dotfiles
cd ~/.dotfiles
```

```
just full-auto      # base packages, languages, CLI tools, fish, yazi,
fastfetch
just full-auto-gui  # the above + kitty + the niri/waybar graphical stack
```

full-auto chains: packages → dev-setup → cli-tools → fish-shell → helix-editor → yazi-file-manager → fastfetch. The packages step is interactive – it pauses so you can review `/etc/xdg/reflector/reflector.conf` and add the Chaotic AUR include to `/etc/pacman.conf`.

If you are not running `full-auto`, run `just packages` first: it sets up Flatpak/Flathub, reflector mirrors, the Chaotic AUR repository, and the `paru` AUR helper that every other recipe uses.

## The stow model

Configuration is deployed as symlinks with **GNU Stow** (`--dotfiles` mode: `dot-foo` in the repo becomes `~/.foo`).

- **Base packages** (stowed on every machine): `home git idea scripts terminal lazyvim vim systemd niri`
- **Host overlays** in `hosts/<hostname>/` (stowed only on the matching machine): `display layout, thermal sensors, machine-specific tools`. The two hosts are `daisy` (AMD) and `xps` (Intel).

```
just stow          # symlink base + this host's overlay (auto-detected)
just stow-check    # dry-run: show what would change, touch nothing
just unstow        # remove all symlinks (host overlay first, then base)
```

`--no-folding` is used so base and host overlays can both populate shared directories (`~/.config/mise`, `~/.local/share/applications`) without tree-folding conflicts.

## Services

```
just services-enable
```

Enables the expected `systemd` **user** services (`mpd`, `syncthing` if installed, `solaar`) and wires the `niri` session helpers (`waybar`, `wpaperd`, `mako`, `swayidle`) to start with `niri.service` via `add-wants`. The `swayosd` backend is a **system** unit and gets `sudo systemctl enable` instead.

## Tool versions: pacman + mise

Two deliberate layers:

- **pacman/paru** installs tools available in the repos (`just cli-tools`).
- **mise** pins versions for languages and tools *not* in the repos: shared config in `terminal/dot-config/mise/config.toml`, per-host additions in `hosts/<hostname>/dot-config/mise/conf.d/<hostname>.toml`.

## Day to day

```
just update      # paru -Syu, mise upgrade, fisher, yazi plugins, nvim Lazy sync
just doctor      # health check: tools, login shell, host overlay, services,
symlinks
```

## Docs

This site itself is part of the repo:

```
just docs-setup # one-time: install typst (pandoc + uv come from the repos)
just docs       # regenerate markdown from configs, build HTML site + PDFs
just docs-serve # live-preview the site while editing
```

tools/gen-docs.py (Python stdlib, no deps) re-reads `tmux.conf`, `binds.kdl`, and the justfile and rewrites the generated blocks in the markdown. Zensical (run via `uvx`, nothing to install) builds the HTML site into `site/`; pandoc + Typst render the PDFs into `docs/pdf/`, which the site links directly.

## Justfile reference

All automation in this repo is a just recipe. This page is generated from the live justfile by `tools/gen-docs.py` — run `just docs-update` to refresh it.

Run from the repo root. `just` with no arguments lists everything.

### Top-level

Recipe	Description	Runs
<code>just full-auto</code>	Full auto installation	<code>packages</code> , <code>dev-setup</code> , <code>cli-tools</code> , <code>fish-shell</code> , <code>helix-editor</code> , <code>yazi-file-</code> <code>manager</code> , <code>fastfetch</code>
<code>just full-auto-gui</code>	Fuller auto (graphical) in- stallation	<code>full-auto</code> , <code>kitty-</code> <code>terminal</code> , <code>niri-window-</code> <code>manager</code>
<code>just packages</code>	(Run first unless running <code>full-auto</code> ) Setup Flatpak, <code>pacman</code> mirrors, Chaotic AUR, and <code>paru</code> AUR helper	
<code>just systemd-niri-config</code>	Enables the <code>systemd</code> ser- vices for some essential <code>niri</code> helpers	<code>systemd-niri-config-</code> <code>install</code> , <code>wpaper-reload</code> , <code>mako-reload</code> , <code>waybar-</code> <code>reload</code> , <code>swayidle-reload</code>
<code>just systemd-niri-</code> <code>config-install</code>		

### Install: essentials

Recipe	Description	Runs
<code>just cli-tools</code>	Essential CLI tools from the pacman repos	
<code>just dev-setup</code>	Programming languages, runtimes, toolchains, and git	
<code>just fish-shell</code>	fish shell and plugins	
<code>just helix-editor</code>		
<code>just yazi-file-manager</code>	Yazi terminal file manager and plugins	

### Install: graphical stack

Recipe	Description	Runs
<code>just kitty-terminal</code>		
<code>just niri-window-manager</code>	Install niri and tools used with it	

### Install: optional extras

Recipe	Description	Runs
<code>just fastfetch</code>		
<code>just syncthing-file-sync</code>		

### Stow (symlink management)

Recipe	Description	Runs
<code>just stow</code>	Stow base packages + this host's overlay (hosts/\$host-name/), auto-detected	
<code>just stow-check</code>	Dry-run the stow above (shows what would change, touches nothing)	
<code>just unstow</code>	Remove all symlinks created by stow (host overlay first, then base)	

### Services

Recipe	Description	Runs
<code>just services-enable</code>	Enable + start the user services this setup expects	

## niri: reload helpers

Recipe	Description	Runs
<code>just mako-reload</code>		
<code>just swayidle-reload</code>		
<code>just waybar-reload</code>		
<code>just wpaper-reload</code>		

## Maintenance

Recipe	Description	Runs
<code>just doctor</code>	Health check: tools present, shell, host overlay, services, mise, key symlinks	
<code>just update</code>	Bring an existing machine fully up to date (each step best-effort)	

## Docs (this site)

Recipe	Description	Runs
<code>just docs</code>	Full docs build: regenerate markdown, then PDFs + cheatsheets, then the HTML site	<code>docs-pdf,</code> <code>docs-cheatsheets,</code> <code>docs-html</code>
<code>just docs-cheatsheets</code>	Landscape cheatsheet PDFs, Catppuccin Latte (print) + Mocha (dark) variants	<code>docs-update</code>
<code>just docs-html</code>	Build the HTML site into site/ (zensical via uvx)	<code>docs-update</code>
<code>just docs-open</code>	Open a PDF (default: the combined handbook; e.g. <code>just docs-open tmux-shortcuts</code> )	<code>docs-pdf</code>
<code>just docs-pdf</code>	Build the PDFs (per-page references + combined handbook) into docs/pdf/	<code>docs-update</code>
<code>just docs-serve</code>	Live-preview the docs site while editing (opens the browser once it's up)	<code>docs-update</code>
<code>just docs-setup</code>	One-time install of the docs toolchain (pandoc + typst; uv ships with CachyOS)	
<code>just docs-update</code>	Regenerate the generated markdown blocks from <code>tmux.conf</code> , <code>binds.kdl</code> , and this justfile	

## tmux shortcuts

Aliases `mux` / `mux1` open (or re-attach to) the first and second sessions.

## Custom bindings

Generated from `terminal/dot-tmux.conf` by `tools/gen-docs.py` — run `just docs-update` to refresh.

The prefix is `C-a` (`Ctrl+a`).

### After the prefix

Press the prefix first, then the key.

Key	Description	Command
C-a	Jump to the last active window	last-window
Escape	Enter copy mode	copy-mode
p	Paste buffer	paste-buffer
r	Reload ~/.tmux.conf	source-file \${HOME}/.tmux.conf \; display-message "source-file reloaded"
C-l	Send a literal Ctrl+L (clear screen) to the pane	send-keys 'C-l'
=	Split pane <b>horizontally</b> (keeps current path)	split-window -h -c "#{pane_current_path}"
-	Split pane <b>vertically</b> (keeps current path)	split-window -v -c "#{pane_current_path}"
M	Mouse mode <b>ON</b>	set -g mouse on \; display 'Mouse: ON'
m	Mouse mode <b>OFF</b>	set -g mouse off \; display 'Mouse: OFF'
u	Capture pane and pick a URL to open (urlview popup window)	capture-pane \; save-buffer /tmp/tmux-buffer \; new-window -n "urlview"...
C-f	Popup fuzzy <b>file picker</b> (pick-files)	display-popup -E -h 70% -w 70% "/home/rdlu/.local/bin/pick-files"
C-t	Popup <b>terminal</b> in the current pane's path	display-popup -E -h 70% -w 70% -d " #{pane_current_path}"

### No prefix (root table)

Active anywhere, no prefix needed.

Key	Description	Command
F12	Toggle <b>OFF mode</b> for nested/remote tmux — all keys (incl. the prefix) pass through to the pane; status bar turns red	set prefix None \; set key-table off \; set status-style "fg=colour245,..."

### Copy mode (vi keys)

Active while in copy mode.

Key	Description	Command
v	Begin selection (vi-style)	send-keys -X begin-selection
C-v	Toggle rectangle (block) selection	send-keys -X rectangle-toggle
y	Copy selection and leave copy mode	send-keys -X copy-selection-and-cancel

### OFF mode

Active only while OFF mode is toggled on.

Key	Description	Command
F12	Leave OFF mode (restore the local prefix)	set -u prefix \; set -u key-table \; set -u status-style \; set -u wind...

### tmux defaults worth remembering

Key (after prefix)	Action
c	New tab (window)
1 2 3...9	Change to tab #1 to #9
!	Promote pane to tab
,	Rename tab
w	List windows
f	Find window
&	Kill window
.	Move window (prompt #)
:movew + Enter	Move window (unused #)
\$	Rename session

### Plugin bindings (TPM)

Plugin keymaps are defined by the plugins themselves, not in `tmux.conf`, so these tables are maintained by hand. The plugin list lives in the `@tpm_plugins` block of `tmux.conf`. Manage plugins with `Prefix + I` (install) and `Prefix + U` (update all).

## copycat

Key (after prefix)	Action
/	regex search + copy mode
n	jump to the next match in copy mode
N	jump to the previous match
y	copy entire line
Enter	copy highlighted match

## yank

- y – copy selection to the system clipboard

*Tip with mouse support: press y before releasing the mouse.*

## extrakto (fuzzy search)

Key (after prefix)	Action
Tab	fuzzy search + fzf mode
Enter	copy selection inside fzf mode

Requires fzf (paru -S fzf).

## resurrect + continuum

Key (after prefix)	Action
Ctrl+s	save all tabs
Ctrl+r	restore all tabs

Continuum is configured with @continuum-restore on, so sessions also restore automatically when the tmux server starts.

## pain-control (panes)

Splitting:

Key (after prefix)	Action
\	split current pane horizontally
-	split current pane vertically

Navigation:

Key (after prefix)	Action
h C-h	select pane on the <b>left</b>
j C-j	select pane <b>below</b> the current
k C-k	select pane <b>above</b> the current
l C-l	select pane on the <b>right</b>

Resizing:

Key (after prefix)	Action
H	resize <b>left</b>
J	resize <b>down</b>
K	resize <b>up</b>
L	resize <b>right</b>

Swapping windows:

Key (after prefix)	Action
<	move one position to the left
>	move one position to the right

### **sessionist (sessions)**

Key (after prefix)	Action
g	prompt for session name and switch to it
C	prompt for creating a new session by name
X	kill current session without detaching tmux
S	switch to the last session
@	promote current pane into a new session

### **sessionx (session picker)**

Key (after prefix)	Action
o	open the sessionx picker (zoxide mode, ~/Projects)

### **open**

Key (in copy mode)	Action
o	open a highlighted selection with the system default program
Ctrl+o	open a highlighted selection with \$EDITOR
S	web-search the selection (DuckDuckGo)

### **thumbs**

Hint-based copy: trigger with Prefix + Space, then press the hint key to copy that text (configured to use wl-copy via OSC52).

## logging

Key (after prefix)	Action
P	start/stop logging
Alt+p	log current screen
Alt+P	save complete history

## fzf

Prefix + F opens the tmux-fzf menu (sessions, windows, panes, commands).

## zellij shortcuts

Key bindings for zellij – configured with `clear-defaults=true`, so everything below is the complete map.

Generated from `terminal/dot-config/zellij/config.kdl` by `tools/gen-docs.py` – run `just docs-update` to refresh.

The default mode is **locked** (keys pass through to the shell). `Ctrl a` enters **tmux mode** – a tmux-style prefix – and `Ctrl g` toggles the regular zellij modes. Most actions return to locked automatically.

## Tmux mode – Ctrl a

Keys	Action
[	→ scroll mode
Ctrl a	Toggle last tab
z f	Fullscreen pane
c t	New tab
s \$ :	Open session-manager
\	New pane right
-	New pane down
p	Previous tab
n	Next tab
1..6	Go to tab 1..6
,	Rename tab
Left h	Focus pane left
Right l	Focus pane right
Down j	Focus pane down
Up k	Focus pane up
.	Rename pane
H	Grow left
L	Grow right
K	Grow up
J	Grow down
e	Embed ⇌ float pane
w	Show/hide floating panes
d	Detach session
Space	Next layout
x	Close pane
/	Search

### Always available (normal & locked)

Keys	Action
Alt left Alt h	Focus pane/tab left
Alt down Alt j	Focus pane down
Alt up Alt k	Focus pane up
Alt right Alt l	Focus pane/tab right
Alt + Alt =	Grow
Alt -	Shrink
Alt f	Show/hide floating panes
Alt i	Move tab left

### Pane mode — p

Keys	Action
left h	Focus pane left
down j	Focus pane down
up k	Focus pane up
right l	Focus pane right
c	Rename pane
d	New pane down
e	Embed ⇌ float pane
f	Fullscreen pane
n	New pane
p	→ normal mode
r	New pane right
w	Show/hide floating panes
x	Close pane
z	Toggle pane frames
tab	Next pane

## Tab mode – t

Keys	Action
left up h k	Previous tab
down right j l	Next tab
1..9	Go to tab 1..9
[	Break pane to tab left
]	Break pane to tab right
b	Break pane into new tab
n	New tab
r	Rename tab
s	Sync input to all panes in tab
t	→ normal mode
x	Close tab
tab	Toggle last tab

## Resize mode – r

Keys	Action
left h	Grow left
down j	Grow down
up k	Grow up
right l	Grow right
+ =	Grow
-	Shrink
H	Shrink left
J	Shrink down
K	Shrink up
L	Shrink right
r	→ normal mode

### Move mode – m

Keys	Action
left h	Move pane left
down j	Move pane down
up k	Move pane up
right l	Move pane right
m	→ normal mode
n tab	Move pane (next)
p	Move pane backwards

### Scroll mode – s

Keys	Action
Alt left Alt h	Focus pane/tab left
Alt down Alt j	Focus pane down
Alt up Alt k	Focus pane up
Alt right Alt l	Focus pane/tab right
e	Edit scrollbar in \$EDITOR
f	Search
s	→ normal mode

### Scrolling keys (scroll & search)

Keys	Action
PageDown right Ctrl f l	Page down
PageUp left Ctrl b h	Page up
down j	Line down
up k	Line up
Ctrl c	Jump to bottom
d	Half page down
u	Half page up

## Search options

Keys	Action
c	Toggle case-sensitivity
n	Next match
o	Toggle whole-word
p	Previous match
w	Toggle wrap

## Session mode — o

Keys	Action
c	Open configuration
d	Detach session
o	→ normal mode
p	Open plugin-manager
w	Open session-manager

## Locked mode (default)

Keys	Action
Ctrl g	→ normal mode
Ctrl a	→ tmux mode

## Mode switches & misc

Keys	Action
Ctrl g Ctrl a enter esc Ctrl c	→ locked mode
Ctrl q	Quit zellij
m	→ move mode
o	→ session mode
t	→ tab mode
s Ctrl c esc	→ scroll mode
p	→ pane mode
r	→ resize mode
enter	→ search mode
esc	Cancel rename · → tab mode
esc	Cancel rename · → pane mode

## niri shortcuts

Key bindings for the niri scrollable-tiling Wayland compositor.

Generated from `niri/dot-config/niri/binds.kdl` by `tools/gen-docs.py` — run `just docs-update` to refresh. For the mouse-warp binds see also the `wl-kbptr` guide.

Mod is the **Super** key. Press `Mod+Shift+/'` to open these docs as a local web app.

### Apps & launchers

Keys	Action
<code>Mod+Shift+Slash</code>	Dotfiles Docs (local)
<code>Mod+T</code>	Open a Terminal: <code>ghostty</code>
<code>Mod+D</code>	Run an Application: <code>fuzzel</code>
<code>Mod+Slash</code>	Run an Application: <code>fuzzel</code>
<code>Super+Alt+L</code>	Lock the Screen: <code>swaylock</code>
<code>Mod+Escape</code>	Menu Power
<code>Mod+Shift+Escape</code>	Menu Just: dotfiles commands
<code>Mod+E</code>	Run <code>ghostty -e yazi</code>
<code>Mod+Period</code>	Emoji Finder
<code>Mod+V</code>	Manage clipboard
<code>Mod+B</code>	Hide Waybar

## Mouse warp (wl-kbptr)

Keys	Action
Mod+G	Mouse Warp: wl-kbptr
Mod+Shift+G	Mouse Warp + Click
Mod+Alt+G	Mouse Warp: coarse (tile only)
Mod+Ctrl+G	Mouse Warp: tile + split
Mod+A	Mouse Warp: other monitor
Mod+Shift+A	Mouse Warp + Click: other monitor
Mod+Alt+A	Mouse Warp: coarse, other monitor
Mod+Ctrl+A	Mouse Warp: tile + split, other monitor
Mod+N	Mouse Warp: bisect then tile
Mod+Shift+N	Mouse Warp + Click: bisect then tile
Mod+M	Mouse Warp: bisect then tile, other monitor
Mod+Shift+M	Mouse Warp + Click: bisect then tile, other monitor
Mod+X	Mouse Warp: detect + click (X-ray)
Mod+Shift+X	Mouse Warp: detect only

## Media, volume & brightness

Keys	Action
XF86AudioRaiseVolume	Run swayosd-client --output-volume raise <i>(works on lock screen)</i>
XF86AudioLowerVolume	Run swayosd-client --output-volume lower <i>(works on lock screen)</i>
XF86AudioMute	Run swayosd-client --output-volume mute-toggle <i>(works on lock screen)</i>
XF86AudioMicMute	Run swayosd-client --input-volume mute-toggle <i>(works on lock screen)</i>
XF86MonBrightnessUp	Run swayosd-client --brightness +10 <i>(works on lock screen)</i>
XF86MonBrightnessDown	Run swayosd-client --brightness -10 <i>(works on lock screen)</i>
XF86AudioNext	Run playerctl next <i>(works on lock screen)</i>
XF86AudioPause	Run playerctl play-pause <i>(works on lock screen)</i>
XF86AudioPlay	Run playerctl play-pause <i>(works on lock screen)</i>
XF86AudioPrev	Run playerctl previous <i>(works on lock screen)</i>

## Focus

Keys	Action
Mod+Left	Focus column left
Mod+Down	Focus window down
Mod+Up	Focus window up
Mod+Right	Focus column right
Mod+H	Focus column left
Mod+L	Focus column right
Mod+Home	Focus column first
Mod+End	Focus column last
Mod+WheelScrollRight	Focus column right
Mod+WheelScrollLeft	Focus column left
Mod+Shift+WheelScrollDown	Focus column right
Mod+Shift+WheelScrollUp	Focus column left

## Moving windows & columns

Keys	Action
Mod+Ctrl+Left	Move column left
Mod+Ctrl+Down	Move window down
Mod+Ctrl+Up	Move window up
Mod+Ctrl+Right	Move column right
Mod+Ctrl+H	Move column left
Mod+Ctrl+L	Move column right
Mod+Ctrl+Home	Move column to first
Mod+Ctrl+End	Move column to last
Mod+Ctrl+WheelScrollRight	Move column right
Mod+Ctrl+WheelScrollLeft	Move column left
Mod+Ctrl+Shift+WheelScrollDown	Move column right
Mod+Ctrl+Shift+WheelScrollUp	Move column left
Mod+BracketLeft	Consume or expel window left
Mod+BracketRight	Consume or expel window right
Mod+Comma	Consume window into column
Mod+Shift+Comma	Expel window from column

## Workspaces

Keys	Action
Mod+J	Focus window or workspace down
Mod+K	Focus window or workspace up
Mod+Ctrl+J	Move window down or to workspace down
Mod+Ctrl+K	Move window up or to workspace up
Mod+Shift+Ctrl+J	Move Workspace to Monitor Left
Mod+Shift+Ctrl+K	Move Workspace to Monitor Right
Mod+Page_Down	Focus workspace down
Mod+Page_Up	Focus workspace up
Mod+U	Focus workspace down
Mod+I	Focus workspace up
Mod+Ctrl+Page_Down	Move column to workspace down
Mod+Ctrl+Page_Up	Move column to workspace up
Mod+Ctrl+U	Move column to workspace down
Mod+Ctrl+I	Move column to workspace up
Mod+Shift+Page_Down	Move workspace down
Mod+Shift+Page_Up	Move workspace up
Mod+Shift+U	Move workspace down
Mod+Shift+I	Move workspace up
Mod+WheelScrollDown	Focus workspace down ( <i>cooldown 150ms</i> )
Mod+WheelScrollUp	Focus workspace up ( <i>cooldown 150ms</i> )
Mod+Ctrl+WheelScrollDown	Move column to workspace down ( <i>cooldown 150ms</i> )
Mod+Ctrl+WheelScrollUp	Move column to workspace up ( <i>cooldown 150ms</i> )
Mod+1	Focus workspace 1
Mod+2	Focus workspace 2
Mod+3	Focus workspace 3
Mod+4	Focus workspace 4
Mod+5	Focus workspace 5
Mod+6	Focus workspace 6
Mod+7	Focus workspace 7
Mod+8	Focus workspace 8
Mod+9	Focus workspace 9
Mod+Ctrl+1	Move column to workspace 1
<b>Mod+Ctrl+0</b>	<b>Move column to workspace 0</b>

## Monitors

Keys	Action
Mod+Shift+Left	Focus monitor left
Mod+Shift+Down	Focus monitor down
Mod+Shift+Up	Focus monitor up
Mod+Shift+Right	Focus monitor right
Mod+Shift+H	Focus monitor left
Mod+Shift+J	Focus monitor down
Mod+Shift+K	Focus monitor up
Mod+Shift+L	Focus monitor right
Mod+Shift+Ctrl+Left	Move column to monitor left
Mod+Shift+Ctrl+Down	Move column to monitor down
Mod+Shift+Ctrl+Up	Move column to monitor up
Mod+Shift+Ctrl+Right	Move column to monitor right
Mod+Shift+Ctrl+H	Move column to monitor left
Mod+Shift+Ctrl+L	Move column to monitor right

## Layout & sizing

Keys	Action
Mod+R	Switch preset column width
Mod+Shift+R	Switch preset window height
Mod+Ctrl+R	Reset window height
Mod+Z	Maximize column
Mod+Shift+Z	Fullscreen window
Mod+Ctrl+Z	Expand column to available width
Mod+C	Center column
Mod+Ctrl+C	Center visible columns
Mod+Minus	Set column width -10%
Mod+Equal	Set column width +10%
Mod+Shift+Minus	Set window height -10%
Mod+Shift+Equal	Set window height +10%
Mod+F	Toggle window floating
Mod+Shift+F	Switch focus between floating and tiling
Mod+W	Toggle column tabbed display
Mod+Space	Switch layout next
Mod+Shift+Space	Switch layout prev

## Screenshots & screencasting

Keys	Action
Print	Screenshot
Ctrl+Print	Screenshot screen
Alt+Print	Screenshot window
Mod+P	Set dynamic cast window
Mod+Ctrl+P	Set dynamic cast monitor
Mod+Alt+P	Clear dynamic cast target

## Session & misc

Keys	Action
Mod+O	Toggle overview ( <i>no key-repeat</i> )
Mod+Q	Close window
Mod+Alt+Escape	Toggle keyboard shortcuts inhibit ( <i>never inhibited</i> )
Mod+Shift+E	Quit
Ctrl+Alt+Delete	Quit
Mod+Shift+P	Power off monitors

## Shell shortcuts (fish)

Interactive-shell bindings: custom bind lines from this repo, plus the bindings injected by atuin, fzf.fish, and fish itself.

### Custom bindings

Generated from the bind lines in fish/ by tools/gen-docs.py — run just docs-update to refresh.

Keys	Action	Defined in
Alt+f	Open the yazi file manager	fish/ aliases.d/files.fish
Ctrl+h	Delete the word left of the cursor	fish/ aliases.d/keybinds.fish
Ctrl+e	Neovim config picker (default / nvim-lazy / nvim-full)	fish/conf.d/nvim-switcher.fish

### thefuck — fix the last command

Bound inside fish\_user\_key\_bindings (fish/conf.d/thefuck.fish), so they only appear in interactive shells.

Keys	Action
Esc Esc or Ctrl+f	Correct the previous command (thefuck)
fuck (typed)	Same, as a function

### History — atuin

atuin init fish --disable-up-arrow (in config.fish), so the up arrow keeps plain fish history.

Keys	Action
Ctrl+r	Fuzzy-search shell history (atuin)
?	Ask atuin AI (on an empty prompt)

## fzf.fish pickers

Keys	Action	Alias
Ctrl+Alt+f	Search directory (files; Ctrl+d/Ctrl+f reload dirs/ files)	fsd
Ctrl+Alt+l	Git log	fsgl
Ctrl+Alt+s	Git status (changed files)	
Ctrl+Alt+p	Processes	fsp
Ctrl+v	Shell variables	

## fish built-ins worth remembering

Keys	Action
Alt+e	Edit the command line in \$EDITOR
Alt+s	Prepend sudo to the current (or last) command
Alt+- Alt+→	Previous / next working directory (prevd / nextd)
Alt+↑ Alt+↓	Recall arguments from previous commands
Alt+l	List the directory of the token under the cursor
Alt+p	Append &  less (paginate output)
Alt+h	Man page for the current command
Alt+w	One-line description of the current command
Ctrl+w	Delete one path component to the left
Ctrl+z	Undo last edit

## Handy abbreviations & functions

The abbreviation-tips plugin reminds you of these when you type the long form.

Type	Expands to / does
<code>mux / mux1</code>	open or re-attach tmux session <code>mux0 / mux1</code>
<code>nvims</code>	fzf picker for neovim configs (default / <code>nvim-lazy / nvim-full</code> )
<code>nv / lvi</code>	neovim with the light / lazy config
<code>fsd / fsgl / fsp</code>	fzf search: directory / git log / processes
<code>z / zi</code>	zoxide: jump to dir / interactive picker
<code>b</code>	back to the previous directory and list it
<code>mcd</code>	<code>mkdir -p + cd</code> into it
<code>cpv</code>	copy with progress bar ( <code>rsync</code> )

## wl-kbptr

Keyboard-driven mouse pointer for Wayland. Replaces mouseless, which is X11-shaped and breaks on niri.

Upstream: <https://github.com/moverest/wl-kbptr>

### Install

Bundled with the niri install target:

```
just niri-window-manager
```

Or standalone:

```
yay -S wl-kbptr jq
```

`jq` is required by the `wl-kbptr-other-output` helper (see Multi-monitor notes). Config lives at `~/.config/wl-kbptr/config`, stowed from `niri/dot-config/wl-kbptr/config` in this repo.

### Binds

	Current monitor	Other monitor
<b>default</b> (tile,bisect)	Mod+G	Mod+A
<b>+ click</b>	Mod+Shift+G	Mod+Shift+A
<b>coarse</b> (tile only)	Mod+Alt+G	Mod+Alt+A
<b>tile + split</b>	Mod+Ctrl+G	Mod+Ctrl+A
<b>bisect</b> → <b>tile</b>	Mod+N	Mod+M
<b>bisect</b> → <b>tile + click</b>	Mod+Shift+N	Mod+Shift+M
<b>detect + click</b> (X-ray)	Mod+X	—
<b>detect only</b>	Mod+Shift+X	—

The Mod+N/Mod+M families coarse-aim first, then refine on a grid: press the chord, bisect into the rough quadrant with `a/s/d/f/z/x/c/v` (positional left hand), recurse as deep as you want, then press **Enter** to hand the current area off to tile. Tile then shows labeled cells over that area

– type the label, cursor warps. Different mental model from Mod+G (“scan grid, type label”): “point first, label second”.

Other-monitor binds run `~/local/bin/wl-kbptr-other-output` (stowed from `niri/dot-local/bin/wl-kbptr-other-output`). The helper picks the non-focused output, calls `niri msg action focus-monitor` to flip keyboard focus there, then runs `wl-kbptr -O <name>`. Extra args are forwarded so each variant passes its own `-o modes=...`

## Mid-stream click keys

In any mode, pressing **j/k/l** commits the current cursor position and clicks. No need to chain click mode for occasional clicks.

Key	Click
j	left
k	right
l	middle

## Modes

Mode	What it does
tile	Grid of labeled cells. Type the label → cursor jumps to cell center.
bisect	Splits area into 2x4 (or 2x2 if square). One keypress per cell to refine.
split	Crosshair-style refinement; arrow keys (or wasd/h) shrink the area toward that side.
click	Commits position with a click. Default left; override with <code>-o mode_click.button=right</code> .
floating	Reads <code>WxH+X+Y</code> rectangles from stdin. Skip unless you script your targets.

Chains are left-to-right: each mode hands the chosen area to the next.

## Cell-pick keys (home\_row\_keys)

`home_row_keys` is exactly 11 characters: first 8 = bisect cells, last 3 = click buttons (left/right/middle).

Our config is `asdfzxcvjl`:

```
keyboard      screen cells
a s d f       0 1 2 3   (top)
z x c v       4 5 6 7   (bottom)

j = left click
k = right click
l = middle click
```

Left hand drives bisect, right hand drives clicks. Symmetric, all stays on the home row.

## Universal keys

Key	Action
Enter	Commit current position; advance to next mode in chain
Backspace	Undo last keypress (works across mode boundaries)
Escape	Cancel – no warp, no click

## Useful CLI flags

Flag	Use case
-0 <NAME>	Target a specific output (DP-2, eDP-1, ...).
-r WxH+X+Y	Restrict overlay to a region (e.g. inside a single window).
-o KEY=VAL	Override config inline (-o modes=tile,bisect,click).
-p	Print result, don't warp or click. Test mode.
-c FILE	Load alternate config file.

## Multi-monitor notes

- Default behavior (no -0): wl-kbptr passes NULL to layer-shell; niri assigns the overlay to the focused output. So Mod+G always uses the monitor your keyboard focus is on.
- Mod+M family targets the *other* monitor via the helper script. niri only routes keyboard input to a layer-surface on the focused output, so the helper calls `niri msg action focus-monitor <name>` before launching wl-kbptr. Without that, the overlay appears but typing keeps going to the previously-focused window.
- 3+ monitors: the helper picks an arbitrary non-focused one. For deterministic ordering, edit `wl-kbptr-other-output`.

## Gotchas

- `~/.local/bin` is not in niri's PATH. Bind helper scripts with the full `~/.local/bin/<name>` path. Bare command names silently no-op.
- **Split mode + our home\_row\_keys.** j/k/l are click buttons, so they don't navigate in split. Use arrow keys, or w/a/s/h for up/left/down/left fallback. See issue #42.
- **Maintainer comment in #42 misstates click order.** The thread says i, j, k = right/left/middle, but the v0.4.1 source has indices 8/9/10 as LEFT/RIGHT/MIDDLE. Trust the source.

## Files

```
docs/wl-kbptr.md           # this file
niri/dot-config/wl-kbptr/config # home_row_keys + default mode chain
niri/dot-config/niri/binds.kdl  # Mod+G and Mod+M bind families
niri/dot-local/bin/wl-kbptr-other-output # helper for the Mod+M family
```