

HERDR QUICK REFERENCE

Workspaces, panes, agent states, CLI automation (v0.9.x)

Mental Model

Layers

- session** Background server namespace; owns every process
- workspace** One per repo or task; sidebar rolls up agent state
- tab** A layout inside a workspace (agents, logs, server)
- pane** A real terminal; split right or down, survives detach
- agent** A coding agent Herdr recognizes inside a pane

Detaching the client leaves the server and all panes running

Agent States

- working** Agent is busy on a turn
- blocked** Waiting on an approval prompt or a question
- done** Finished, and you have not looked yet
- idle** Finished or waiting, already seen
- unknown** Agent present, state unclear; not proof it finished

idle and done both mean the agent will accept a new prompt

Input Modes

- terminal** Keys go straight to the focused pane
- prefix** After Ctrl-b, the next key is a Herdr action
- navigate** Persistent workspace navigation surface
- copy** Keyboard scrollbar, search and select (Prefix [])

Getting Started

Install & Launch

```
curl -fsSL https://herdr.dev/install.sh | sh
cd ~/projects/app
herdr # launch or attach
claude # any agent, auto-detected
herdr update # self-update
herdr status # client vs server version
```

Agent Integrations

```
herdr integration install claude
herdr integration install codex
herdr integration status --outdated-only
```

Integrations report exact state and let agents resume after a server restart

Mouse First

Everything is clickable: panes, tabs, workspaces, agent rows. Drag borders to resize, right-click for split and tab menus, drag-select to copy. Keys are an optional layer on top.

Key Bindings

Learn These First

- Prefix ?** Show every active binding (/ to filter)
- Prefix c** New tab
- Prefix v** Split: new pane to the right
- Prefix -** Split: new pane below
- Prefix h/j/k/l** Focus pane left / down / up / right
- Prefix w** Workspace navigation
- Prefix q** Detach; everything keeps running

Default prefix is Ctrl-b. Detach is q, not tmux's d

Panes

- Prefix z** Zoom (fullscreen) focused pane
- Prefix x** Close pane
- Prefix Shift-h/j/k/l** Swap pane with its neighbor
- Prefix r** Resize mode
- Prefix Tab** Cycle to next pane
- Prefix Shift-p** Rename pane
- Prefix e** Open scrollbar in an editor

Tabs & Workspaces

- Prefix n / p** Next / previous tab
- Prefix 1..9** Jump to tab by number
- Prefix Shift-t / Shift-x** Rename / close tab
- Prefix Shift-n** New workspace
- Prefix Shift-g** New workspace on a git worktree
- Prefix Shift-w / Shift-d** Rename / close workspace
- Prefix g** Goto picker: every agent and terminal

- Prefix o** Jump to the latest notification's pane
- Prefix b** Toggle sidebar
- Prefix s** Settings

In the Goto picker, b/w/i/d filter blocked/working/idle/done; / searches

Copy Mode (Prefix [])

- h/j/k/l w/b/e** Move by character or word
- Ctrl-u / Ctrl-d** Half page up / down
- / ? n N** Search forward / back, repeat
- v or Space** Start selection
- y or Enter** Copy selection and exit
- q or Esc** Leave without copying

The pane keeps running while you scroll back

Sessions & Remote

Persistence

```
herdr # reattach
herdr session list
herdr session attach work # isolated server
herdr session stop work
herdr server stop # kills ALL panes
```

Over SSH

```
herdr --remote workbox # local UI, remote panes
herdr --remote workbox --session agents
herdr machine add workbox --label box
herdr --machine box agent list
```

Workspaces as Worktrees

```
herdr worktree create --branch fix/login --base main
herdr worktree list
herdr worktree remove --workspace w3
```

Scripting Panes

IDs & Context

IDs look like w1 (workspace), w1:t1 (tab), w1:p1 (pane). Inside a Herdr pane, HERDR_ENV=1 and HERDR_PANE_ID are set. Commands return JSON; read new IDs from the response, never guess them from sidebar order.

Discover

```
herdr workspace list
herdr tab list --workspace "$HERDR_WORKSPACE_ID"
herdr pane list --workspace "$HERDR_WORKSPACE_ID"
herdr pane layout --current
```

Run a Command Next Door

```
id=$(herdr pane split --current \
  --direction right --no-focus \
  | jq -r '.result.pane.pane_id')
herdr pane run "$id" "npm test"
herdr pane wait-output "$id" \
  --match "passed" --timeout 120000
herdr pane read "$id" --lines 80 \
  --source recent-unwrapped
```

Pane Commands

- pane run ID CMD** Type a command and press Enter in one step
- pane send-text ID TEXT** Type literal text, no Enter
- pane send-keys ID ctrl+c** Send named keys
- pane wait-output --regex RE** Block until output matches (Rust regex)
- pane zoom ID --toggle** Zoom a pane from a script
- pane move ID --new-tab** Move a pane; it gets a new ID
- notification show TITLE** Pop a toast, optionally with --sound done

Orchestrating Agents

Start, Prompt, Collect

```
herdr agent start reviewer --kind codex --pane "$id"
herdr agent prompt reviewer "Review the diff" \
  --wait --timeout 300000
herdr agent read reviewer --lines 120
herdr agent wait reviewer --until blocked
herdr agent send-keys reviewer esc
```

Agent Commands

- agent list** Every live agent with its state
- agent get NAME** One agent's pane, kind and state
- agent focus NAME** Jump the UI to that agent (marks it seen)
- agent attach NAME** Open just that terminal, no Herdr UI
- agent explain NAME** Why detection chose this state

Targets are a unique agent name or the pane ID hosting it

Pane vs Agent Commands

CHOICE	USE WHEN	WATCH FOR
pane	Shells, tests, servers	Blind to agent state
agent	A detected coding agent	Refuses blocked agents

Rules of Thumb

agent start needs an existing pane sitting at a shell prompt; it never creates layout. Use --no-focus for background work. After a timeout, read the agent before re-sending: the prompt may already have landed.

Configuration

~/config/herdr/config.toml

```
[keys]
prefix = "ctrl+a"
focus_pane_left = ["prefix+h", "ctrl+alt+h"]
[theme]
name = "tokyo-night"
[ui.toast]
delivery = "system" # or herdr / terminal
```

Custom Popup Command

```
[[keys.command]]
key = "prefix+alt+g" # or "pane" / "shell"
type = "popup"
command = "lazygit"
width = "80%"
height = "80%"
```

Config Commands

- herdr --default-config** Print every option with comments
- herdr config check** Validate config.toml
- Prefix Shift-r** Reload config in the running server
- herdr config reset-keys** Back up config, drop custom bindings

ctrl+alt chords are the safest prefix-free bindings across terminals