

Printable setup guide. Covers the out-of-box path for turning Harness into a tmux-style terminal: install, Full Terminal mode, prefix shortcuts, panes, tabs, copy mode, attach-window, scripting, shell integration, and agent notifications.

Harness tmux-style capabilities

This guide is the printable setup and usage path for Harness's tmux-style features: sessions, tabs, panes, copy mode, paste buffers, attach from another terminal, key bindings, shell integration, and agent notifications.

Harness is not tmux and does not wrap tmux. The terminal engine, daemon, session model, compositor, and CLI are first-party Harness components. The naming is familiar where it helps: a Harness tab is the same idea as a tmux window, and a Harness pane is a split terminal.

1. Five-minute setup

Use this path for a fresh install.

1. Install Harness.

Download the DMG, drag `Harness.app` to Applications, and launch it once. Harness starts the background daemon automatically.

2. Pick the tmux-style experience.

Open **Settings > Terminal > Experience** and choose **Full Terminal** or **Agent Workspace**.

New installs can start in **Plain Terminal**, which intentionally hides the prefix layer and status line. Plain mode is a normal terminal. Full Terminal turns on the prefix key, status line, copy mode, paste buffers, panes, and command prompt.

3. Install the CLI.

From the app bundle:

```
/Applications/Harness.app/Contents/MacOS/harness-cli install
```

If you built from source:

```
.build/release/harness-cli install
```

Add the printed bin path to your shell profile if needed:

```
export PATH="$HOME/Library/Application Support/Harness/bin:$PATH"
```

4. Verify the daemon and CLI.

```
harness-cli ping
harness-cli list-sessions
harness-cli list-windows
```

A healthy install prints `pong` and shows your current sessions/tabs.

5. Install shell integration.

```
harness-cli install-shell-integration
```

Open a new Harness pane after installing. This enables prompt marks, copy-mode prompt jumps, and the command success/failure gutter.

6. Optional: install agent hooks.

```
harness-cli install-hooks claude-code
harness-cli install-hooks codex
```

Use the agent you run: `claude-code`, `codex`, `cursor`, `pi`, `hermes`, or `openclaw`. Harness also detects several agents without hooks.

2. Mental model

Harness term	tmux-style idea	What it means
Workspace	Top-level project group	A named group of sessions.
Session	Session	A sidebar row with its own tabs.
Tab	Window	One terminal tab with a split tree.
Pane	Pane	One PTY-backed terminal inside a tab.
Surface	PTY identity	The daemon-owned terminal behind a pane.

The daemon owns session state and every PTY. The app and `harness-cli` are clients. That is why shells keep running after the app quits, why another Harness window can reattach, and why `harness-cli attach-window` can render a full split layout in a plain terminal.

3. Prefix key

Default prefix: **Ctrl-A**.

When this guide says `prefix %`, press `Ctrl-A`, release it, then press `%`.

Key	Action
<code>prefix ?</code>	Show the live cheatsheet for your current bindings.
<code>prefix :</code>	Open the command prompt.
<code>prefix d</code>	Detach the current client.
<code>prefix [</code>	Enter copy mode.

Change the prefix in **Settings > Keys**. If the prefix does nothing, check that the app is in **Full Terminal** or **Agent Workspace**, or that `tmuxControlsEnabled` is on.

4. Panes and layouts

Shortcut	What it does
prefix %	Split side-by-side, new pane on the right.
prefix "	Split top/bottom, new pane below.
prefix Left/Right/Up/Down	Move focus to the neighboring pane.
prefix o	Cycle to the next pane.
prefix ;	Cycle to the previous pane.
prefix l	Jump to the last active pane.
prefix z	Zoom or unzoom the active pane.
prefix x	Kill the active pane.
prefix Shift-Left/Right/Up/Down	Resize the pane. Hold under prefix to repeat.
prefix q	Show pane numbers, then press a number to jump.
prefix Space	Cycle layout presets.
prefix S	Toggle synchronized input to every pane in the tab.

macOS shortcuts also work:

Shortcut	Action
Cmd-D	Split side-by-side.
Cmd-Shift-D	Split top/bottom.
Cmd-W	Close the active tab.

Layout commands:

```
:select-layout tiled
:select-layout even-horizontal
:select-layout even-vertical
:select-layout main-horizontal
:select-layout main-vertical
```

From the shell:

```
harness-cli new-split --tab <tab-uuid> --direction horizontal
harness-cli resize-pane --pane <pane-uuid> --dir L --amount 5
harness-cli zoom-pane --pane <pane-uuid>
```

5. Tabs, sessions, and workspaces

Harness tabs are tmux-style windows. Sessions live in the sidebar and remain visible instead of being hidden behind an attach prompt.

Shortcut	What it does
prefix c	New tab.
prefix n	Next tab.
prefix p	Previous tab.
prefix ,	Rename current tab.
Cmd-T	New tab.
Cmd-1 through Cmd-9	Jump to tab 1 through 9.
Cmd-Shift-[/ Cmd-Shift-]	Previous / next tab.
Cmd-Shift-N	New workspace.

Useful CLI commands:

```
harness-cli new-workspace --name api
harness-cli new-session --workspace Default --cwd ~/Code/project
harness-cli new-tab --workspace Default --cwd ~/Code/project
harness-cli rename-session --session <uuid> --name backend
harness-cli list-sessions
harness-cli list-windows
harness-cli list-panes
```

Persistence:

- Full and Agent modes are designed for persistent work.
- A session survives a clean quit when global keep-on-quit is on or the session is pinned.
- Pin or unpin a session from the sidebar context menu, or use:

```
harness-cli promote-session --session <uuid>
harness-cli demote-session --session <uuid>
```

6. Copy mode, search, and paste buffers

Enter copy mode with prefix [.

Key	Action
<code>h / j / k / l</code>	Move left/down/up/right.
<code>0 / \$</code>	Start / end of line.
<code>w / b</code>	Next / previous word.
<code>g / G</code>	Top / bottom of history.
<code>PageUp / PageDown</code>	Page through history.
<code>C-u / C-d</code>	Half-page up / down.
<code>[/]</code>	Previous / next shell prompt. Requires shell integration.
<code>v</code>	Start character selection.
<code>V</code>	Start line selection.
<code>C-v</code>	Start rectangle selection.
<code>/</code> then text	Search forward.
<code>?</code> then text	Search backward.
<code>n / N</code>	Next / previous search result.
<code>y</code> or <code>Enter</code>	Yank to clipboard and paste buffer, then exit.
<code>p</code>	Paste latest buffer into the pane.
<code>q</code> or <code>Escape</code>	Exit copy mode.

Paste buffer commands:

```
harness-cli set-buffer --data "deploy staging"
harness-cli list-buffers
harness-cli show-buffer
harness-cli paste-buffer --surface "$HARNESS_SURFACE" --bracketed
harness-cli save-buffer notes.txt
harness-cli load-buffer notes.txt
```

7. Attach from another terminal or over SSH

Install the CLI on the machine that is running Harness, then use:

```
harness-cli attach-window
harness-cli attach-window --session work
harness-cli attach-window --tab <tab-uuid>
```

`attach-window` renders the whole tab: splits, borders, status line, active pane, copy mode, and mouse support. It works in a plain terminal because the Harness client composites the daemon-owned pane grids.

Inside `attach-window`:

Key	Action
prefix % / prefix "	Split.
prefix h/j/k/l	Select pane.
prefix o / prefix ;	Cycle pane.
prefix z	Zoom.
prefix x	Kill pane.
prefix c	New tab.
prefix n / prefix p	Next / previous tab.
prefix d	Detach the attached client.
prefix [	Copy mode.

Detach keys default to `Ctrl-A d`. Override them:

```
harness-cli attach-window --detach-keys "C-b d"
```

Single-pane attach is also available:

```
harness-cli attach --surface <surface-uuid>
```

8. Command prompt and scripting

Open the prompt with `prefix :` or `Cmd-;`. It accepts the same command grammar as key bindings and hooks:

```
:split-window -h
:split-window -v
:select-layout tiled
:display-message "#{session_name}:#{window_index}.#{pane_index}"
:bind-key C-x x kill-pane
:bind-key -T copy-mode y copy-mode -X copy-selection-and-cancel
```

From a shell:

```
harness-cli send-keys --surface "$HARNESS_SURFACE" --keys "git status Enter"
harness-cli capture-pane --surface "$HARNESS_SURFACE"
harness-cli pipe-pane --surface "$HARNESS_SURFACE" "tee pane.log"
harness-cli display-message '#{cwd_basename}'
harness-cli wait-for -S build-ready
```

9. Key binding customization

Bindings persist in:

```
~/Library/Application Support/Harness/keybindings.json
```

Examples:

```
# Bind Ctrl-X q to detach.
harness-cli bind-key C-x q detach-client

# Move kill-pane off x and onto Ctrl-X x.
harness-cli unbind-key x
harness-cli bind-key C-x x kill-pane

# Bind a multi-command workflow.
harness-cli bind-key C-x s "split-window -h ; copy-mode"

# Copy-mode binding.
harness-cli bind-key -T copy-mode Y "copy-mode -X copy-selection-and-cancel"

# List current bindings.
harness-cli list-keys
harness-cli list-keys -T copy-mode
```

Key syntax:

Form	Meaning
C-a	Control-A.
M-x	Option/Alt/Meta-X.
S-Left	Shift-Left.
Cmd-,	Command-comma.
PageUp, Enter, Escape, F1	Named keys.

10. Status line, mouse, and options

Harness supports tmux-style options through `set-option` and `show-options`.

```
harness-cli show-options -g
harness-cli set-option -g status on
harness-cli set-option -g mouse on
harness-cli set-option -g mode-keys vi
harness-cli set-option -g history-limit 20000
harness-cli set-option -g base-index 1
harness-cli set-option -g pane-base-index 1
```

Common options:

Option	Default	What it controls
<code>status</code>	<code>on</code>	Bottom status line.
<code>status-left</code> / <code>status-right</code>	<code>built in</code>	Status content.
<code>mouse</code>	<code>on</code>	Mouse reporting and pane-click selection.
<code>mode-keys</code>	<code>vi</code>	Copy-mode key style.
<code>set-clipboard</code>	<code>on</code>	Yank to macOS pasteboard.
<code>history-limit</code>	<code>10000</code>	Scrollback cap.
<code>base-index</code>	<code>0</code>	First tab index.
<code>pane-base-index</code>	<code>0</code>	First pane index.

11. Shell integration

Shell integration adds OSC 133 prompt marks. It is optional, but it makes the tmux-style workflow much better.

Install:

```
harness-cli install-shell-integration
harness-cli install-shell-integration all
```

What it enables:

- Prompt gutter: green for exit 0, red for non-zero.
- Copy-mode prompt jumps with `[` and `]`.
- Prompt-aware status and navigation.

The installed snippet is guarded by `$HARNESS`, so it is inert outside a Harness pane. It supports bash, zsh, and fish.

12. Agent notifications

Harness watches coding agents in panes and can surface approval or completion states through desktop notifications, the sidebar bell, pane rings, and `Cmd-Shift-U`.

Install hooks where the agent supports hooks:

```
harness-cli install-hooks claude-code
harness-cli install-hooks codex
harness-cli install-hooks cursor
harness-cli install-hooks pi
harness-cli install-hooks hermes
harness-cli install-hooks openclaw
```

Smoke test from inside a Harness pane:

```
harness-cli notify --surface "$HARNESS_SURFACE" --title Agent --body "Needs approval"
```

Then press `Cmd-Shift-U` to jump to the waiting tab.

If macOS banners do not appear, open **System Settings > Notifications > Harness** and allow notifications. The in-app waiting state still works even when macOS banners are denied.

13. Out-of-box troubleshooting

Symptom	Fix
<code>harness-cli</code> : command not found	Run <code>harness-cli install</code> , then add <code>\$HOME/Library/Application Support/Harness/bin</code> to <code>PATH</code> .
<code>harness-cli ping</code> does not print pong	Launch Harness once, or run the bundled CLI from <code>/Applications/Harness.app/Contents/MacOS/harness-cli</code> .
Prefix does nothing	Choose Full Terminal or Agent Workspace in Settings > Terminal > Experience.
Prompt jumps do nothing	Run <code>harness-cli install-shell-integration</code> , then open a new pane.
Agent hook does not notify	Run <code>harness-cli install-hooks <agent></code> , start the agent in a new Harness pane, and verify <code>\$HARNESS_SURFACE</code> is set.
Desktop banners do not show	Allow Harness in macOS notification settings.
Attach over SSH cannot find sessions	SSH into the same macOS user account that owns the Harness daemon and run <code>harness-cli ping</code> .

14. One-page cheat sheet

Setup

1. Settings > Terminal > Experience > Full Terminal or Agent Workspace
2. `harness-cli install`
3. `harness-cli ping`
4. `harness-cli install-shell-integration`
5. `harness-cli install-hooks <agent>` optional

Prefix

Ctrl-A is the default prefix. `prefix ?` opens the live cheatsheet.

Panes

prefix %	split side-by-side
prefix "	split top/bottom
prefix arrows	focus pane
prefix o/;	next / previous pane
prefix l	last pane
prefix z	zoom
prefix x	kill pane
prefix q	pane numbers
prefix Space	next layout
prefix S	synchronize panes

Tabs

prefix c	new tab
prefix n/p	next / previous tab
prefix ,	rename tab
Cmd-T	new tab
Cmd-1..9	jump to tab

Copy mode

prefix [	enter
h j k l	move
v / V / C-v	select char / line / rectangle
/ or ?	search
n / N	next / previous match
[/]	previous / next prompt
y or Enter	copy selection and exit
p	paste latest buffer
q or Esc	exit

Attach

<code>harness-cli attach-window</code>	
<code>harness-cli attach-window --session work</code>	
prefix d	detach

Command

prefix :	command prompt
Cmd-;	command prompt
Cmd-K	command palette