LINUX · AUGUST 11, 2026

tmux default prefix key bindings: a quick reference for Ctrl-b

Every default tmux prefix binding (Ctrl-b), verified against the tmux 3.7b man page and key-bindings source: sessions, windows, panes, both copy-mode tables, plus an accurate history of what changed in each release.

9 min read · 11 sources

command reference · configuration · copy mode · key bindings · open source · terminal multiplexer · tmux

CONTENTS

How tmux key bindings work 2
Session commands 3
Window commands 4
Pane commands 5
Copy mode: the emacs table 7
Copy mode: the vi table 9
Paste buffers, the command prompt, and diagnostics 12
What changed recently, and keeping this accurate 13
Bottom line 15
Sources 15

tmux is a terminal multiplexer: it runs several terminals inside one screen, keeps them alive after you detach, and lets you reattach from anywhere — the same session, a different machine, days later. Everything you do to tmux itself goes through the **prefix key — `Ctrl-b` by default — followed by a command key**: `C-b c` creates a window, `C-b d` detaches, `C-b %` splits a pane. This is a quick reference to the **default** prefix bindings, verified against the tmux man page [1] [4] and the actual default-bindings source in the tmux tree [2], at the current stable release (**3.7b, 1 July 2026** [6] [8]) and against the in-development 3.8 [3] [11]. Bindings are flagged with the release that introduced them where that is useful; if your installed tmux is older than the flag, the binding will not exist yet. The core of the table — `c`, `d`, `n`, `p`, `l`, `z`, `x`, `o`, `q`, `%`, `"`, `[`, `]` — has been stable for well over a decade, so this reference will stay accurate for a long time.

C-b	3.7b	10 ms	500 ms	50
Default prefix key	Current stable (1 Jul 2026)	Default escape-time (3.5+)	Default repeat-time	Default buffer-limit (paste buffers)

How tmux key bindings work

A tmux key binding is two keystrokes: press the prefix, **release it**, then press the command key. `C-b c` (prefix then `c`) is not the same as `C-b C-c` — release `Ctrl` after the prefix or you send a different key. Pressing the prefix twice, `C-b C-b`, sends a literal `Ctrl-b` through to the program in the pane (`send-prefix`), which is how you talk to a tmux running inside tmux[1][2].

Bindings live in named **key tables**. The important ones are:

- **prefix** — keys pressed after `C-b`; nearly everything in this document.
- **root** — keys pressed with no prefix (the `-n` bindings): the mouse bindings live here.
- **copy-mode** and **copy-mode-vi** — keys active inside copy mode; which one applies depends on the `mode-keys` option (emacs by default, vi if `EDITOR` or `VISUAL` contains "vi")[1][2].

Modifier notation: `C-` is Ctrl, `M-` is Meta (Alt on modern keyboards), `S-` is Shift, and they combine (`C-M-x`). The prefix itself is the `prefix` session option, default `C-b`, with an optional second prefix `prefix2` (unset by default); both can be set to `None` to disable them[1]. The two-prefix arrangement dates to tmux 1.6, which dropped the old single list of prefix keys in favour of "two separate options, prefix and prefix2"[3]. Bindings are added with `bind-key` and removed with `unbind-key`; both take a `-T` flag to choose the table (so `bind -T copy-mode-vi '[' send -X previous-prompt` adds a vi copy-mode binding), `-n` as shorthand for the root table, and `-r` to mark a key repeatable[1].

Several prefix bindings are marked repeatable (`-r` in the source): pane navigation arrows and the resize keys. Once you press one, the binding stays "armed" for **500 ms** (`repeat-time` default; `initial-repeat-time` if set) so you can press the arrow or resize key repeatedly without re-pressing `C-b`. The 3.5 release added `prefix-timeout` (default 500 ms), which cancels a half-typed prefix instead of leaving tmux waiting for a command key that never comes, and cut the default `escape-time` from 500 ms to **10 ms** so Meta and function keys respond snappily[3][7]. Both are worth knowing because they explain two common "why is tmux eating my keystrokes" complaints: a slow `escape-time` makes Meta-key sequences lag, and a short `prefix-timeout` dismisses a prefix you were holding.

Three ways to see the live bindings on your machine:

- `C-b ?` — list all key bindings in a scrollable view (arrows to scroll, `q` to exit). This is the `list-keys -N` note form.
- `C-b /` — describe one key: press a key and tmux shows what it does. This "Describe key binding" binding was added in 3.1 alongside key notes[3].
- From the shell: `tmux lsk -N | less` (all tables with notes), `tmux lsk -T prefix` (prefix table only), `tmux lsk -T copy-mode-vi`[1][2][5].

> CAUTION
>
> A **running tmux server keeps its old behaviour and bindings after an upgrade**. After installing a new tmux, run `tmux kill-server` (or reboot) so every session runs the new binary; otherwise

> newly added bindings will not appear [7] [8]. The 3.7 release notes repeat the standing advice: "As always, fully restart tmux (kill-server) after upgrading so every session runs the new binary" [8].

Configuration lives in `~/.tmux.conf` (also `/etc/tmux.conf`, then `$XDG_CONFIG_HOME/tmux/tmux.conf` on newer builds) and is read once when the server starts. The most common change — and the one every cheat sheet repeats — is swapping the prefix to `Ctrl-a` [1] [5] [10]:

BASH

```
set -g prefix C-a
unbind C-b
bind C-a send-prefix
```

Reload a config without restarting with `C-b : source-file ~/.tmux.conf` (or `tmux source ~/.tmux.conf` from the shell) [9] [10]. Two options are worth knowing alongside the keys: `set -g mouse on` turns on all the mouse bindings described in the pane section (**off by default through 3.7**, on by default in the 3.8 development tree), and `set -g mode-keys vi` switches copy mode to the vi table below [2] [3]. Others that affect everyday feel: `set -g base-index 1` starts window numbering at 1 instead of 0 (many people prefer it, and the shell-prompt tradition of window 0 colliding with the far end of the number row is the usual reason); `set -g renumber-windows on` closes gaps when windows are killed; `set -g history-limit 10000` raises the per-pane scrollback from the default 2000 lines [9] [10]. None of these are defaults, but they are the config lines you will see in every "sane tmux config" guide [9].

Session commands

A session is a named workspace holding one or more windows; it survives disconnects and is the thing you reattach to. Prefix bindings:

KEY	ACTION	COMMAND BEHIND IT
`C-b d`	Detach the current client	`detach-client`
`C-b D`	Choose a client to detach, from a list	`choose-client -Z`
`C-b $`	Rename the current session	`command-prompt -I'#S' { rename-session -- '%%' }`
`C-b s`	Pick a session interactively (tree)	`choose-tree -Zs`
`C-b (` / `C-b )`	Previous / next session	`switch-client -p` / `-n`
`C-b L`	Jump to the last-used session	`switch-client -l`

The same operations from a shell (no prefix needed — these are plain tmux commands):

BASH

```
tmux new -s work          # new session named "work"
tmux new -As work         # attach if it exists, create if not
tmux a -t work            # attach to "work" (tmux attach)
```

```
tmux ls                  # list sessions
tmux rename-session -t work coding
tmux kill-session -t work # kill one session
tmux kill-server         # kill everything, all sessions
```

Sessions are the point of tmux: detach with `C-b d`, close the terminal, reconnect over ssh, and `tmux attach` puts you back exactly where you were[5][9]. The server exits when the last session is killed (`exit-empty`, on by default)[1][11]. New sessions default to names `0`, `1`, `2`…; `new-session -A` ("attach if it exists, create if not") is the handiest flag for a work session you always want to resume[9]. `kill-session -a` kills every session except the current one, and `kill-server` nukes everything including the server process itself[9][10].

The `C-b s` tree (`choose-tree`) is more than a session picker: with `-Zs` it zooms while open, and inside it you can search with `C-s`, tag items with `t`, kill with `x`, and change the sort order with `O`[1]. `C-b D` (choose-client) is the one to remember when you have multiple terminals attached to the same session and want to drop one without leaving the other. `C-b (` and `C-b )` cycle sessions in order, and `C-b L` snaps back to the one you were using before — the session-level twin of `C-b l` for windows and `C-b ;` for panes.

Window commands

Windows are like browser tabs inside a session — each takes the full screen. The status line at the bottom lists them: the current window gets a `*` after its name, the previously selected one a `-`[1][5].

KEY	ACTION	COMMAND BEHIND IT
`C-b c`	Create a new window	`new-window`
`C-b ,`	Rename the current window	`command-prompt -I'#W' { rename-window -- '%%' }`
`C-b n` / `C-b p`	Next / previous window	`next-window` / `previous-window`
`C-b l`	Toggle to the last-used window	`last-window`
`C-b 0`–`C-b 9`	Jump to window 0–9	`select-window -t:=N`
`C-b w`	Pick a window from a list	`choose-tree -Zw`
`C-b f`	Search windows by name/title/content	`find-window -Z`
`C-b '`	Prompt for a window index	`command-prompt { select-window -t ':%%' }`
`C-b .`	Move the window to a new index	`command-prompt { move-window -t '%%' }`
`C-b &`	Kill the current window (asks y/n)	`confirm-before kill-window`
`C-b M-n` / `C-b M-p`	Next / previous window **with an alert**	`next-window -a` / `previous-window -a`
`C-b Tab`	Fuzzy-match a window (3.7)	`switch-mode -wk`
`C-b BTab`	Fuzzy-match a session (3.7)	`switch-mode -sk`

KEY	ACTION	COMMAND BEHIND IT
`C-b <`	Open the window context menu (since 3.0)	`display-menu`

`Tab` / `BTab` (Shift-Tab) are the 3.7 "switch-mode" fast switcher: they open a small floating pane that filters windows (or sessions) as you type, matching terms with the same fuzzy logic as the format `m/z` operator[3][7]. In a session with a dozen windows, `C-b Tab` then two letters beats `C-b w` plus arrow keys. The window menu (`C-b <`, default since 3.0[3]) offers swap-left/right, kill, respawn, mark, rename, new-after and new-at-end, each with a shortcut key shown in brackets — `X` kills, `n` renames, `w` creates a window after the current one[2].

The status-line window flags mean: `*` current window, `-` last window, `#` activity detected (`monitor-activity`), `!` bell, `~` silence, `M` contains the marked pane, `Z` active pane zoomed[1]. When windows overflow the status line, `<` and `>` appear at the ends to show hidden windows[5].

From the shell: `tmux neww -n name` (named window), `tmux swap-window -s 2 -t 1`, `tmux move-window -r` (renumber to close gaps), `tmux link-window -s other:1` (share a window across sessions)[9][10]. `C-b M-n` / `M-p` are the alert keys — they skip straight to the next window that has activity or a bell, which is the fastest way to find the build that just finished or the job that just failed.

Pane commands

Panes are the rectangles a window is split into. This is where most of the muscle memory lives.

KEY	ACTION	COMMAND BEHIND IT
`C-b %`	Split left/right (new pane to the right)	`split-window -h`
`C-b "`	Split top/bottom (new pane below)	`split-window` (defaults to `-v`)
`C-b ↑↓←→`	Move to the pane in that direction	`select-pane -U/-D/-L/-R` (repeatable)
`C-b o`	Cycle to the next pane	`select-pane -t:.+`
`C-b ;`	Toggle to the last-used pane	`last-pane`
`C-b q`	Show pane numbers; press one to jump	`display-panes`
`C-b z`	Toggle zoom (pane fills the window)	`resize-pane -Z`
`C-b x`	Kill the current pane (asks y/n)	`confirm-before kill-pane`
`C-b !`	Break the pane out into its own window	`break-pane`
`C-b {` / `C-b }`	Swap pane with the previous / next	`swap-pane -U` / `-D`
`C-b C-o` / `C-b M-o`	Rotate panes forwards / backwards	`rotate-window` / `rotate-window -D`
`C-b m` / `C-b M`	Mark the pane / clear the mark	`select-pane -m` / `-M`

KEY	ACTION	COMMAND BEHIND IT
`C-b E`	Spread the current pane and neighbours evenly (since 2.7)	`select-layout -E`
`C-b C-↑↓←→`	Resize pane by **1 cell** (repeatable)	`resize-pane -U/-D/-L/-R`
`C-b M-↑↓←→`	Resize pane by **5 cells** (repeatable)	`resize-pane … 5`
`C-b M-1` … `M-7`	Preset layouts (see below)	`select-layout`
`C-b Space`	Cycle to the next preset layout	`next-layout`
`C-b *`	New **floating pane** (3.7)	`new-pane`
`C-b @`	Toggle pane between floating and tiled (3.8, in development)	`break-pane -W` / `join-pane`
`C-b g`	Floating-pane move table (3.8, in development)	`switch-client -Tmove`
`C-b >`	Open the pane context menu (since 3.0)	`display-menu`
`C-b S-↑↓←→`	Scroll the visible part of an oversized window; `C-b DC` (Delete) re-follows the cursor	`refresh-client -U/-D/-L/-R 10`

The preset layouts behind `M-1`…`M-7` are: `M-1` even-horizontal (columns), `M-2` even-vertical (rows), `M-3` main-horizontal (big pane on top), `M-4` main-vertical (big pane left), `M-5` tiled, and — added in 3.5 — `M-6` main-horizontal-mirrored (main pane bottom) and `M-7` main-vertical-mirrored (main pane right)[1][3]. `C-b Space` steps through all of them in order, which is the fastest way to find one you like; `select-layout -o` undoes the last layout change if you overshoot[3].

A few of these deserve detail. `C-b g` switches into a temporary "move" key table for floating panes: `1`–`4` snap the floating pane to the four corners, `0` fills the window, the arrows move it to an edge, `M-1`–`M-4` and `M-arrows` move **and** resize it to half-size, and `,` / `.` open move and move-and-resize menus. It only does anything while a floating pane exists[2][3].

Floating panes (3.7) sit above the tiled layout "like popups but unlike popups are not modal and behave like panes"[3]. In 3.7 they are an early feature — the changelog is candid: "Currently floating panes can only be moved and resized using the mouse"[3]. 3.8 fills in the gaps with keyboard move/resize via `C-b g`, the `@` float/tile toggle, and split-window-inside-floating-pane support[3]. The pane menu (`C-b >`) reflects the floating-pane machinery: it shows Float/Tile (`f`), Move and Move & Resize submenus, plus split, swap, kill, respawn, mark, zoom and paste, with context-sensitive items appearing only when they apply (for example "Search For <word under mouse>" when the mouse is over text)[2].

Mouse behaviour, for the record: the `mouse` option is **off by default through 3.7**, and "The mouse option now defaults to on" in the 3.8 development tree[3]. When it is on, the default root bindings (mostly added in 3.0–3.2) give you: click to select a pane, drag on a border to resize, wheel to scroll (entering copy mode), button 2 (middle click) to paste, double/triple click to copy a word/line, and right-click on a pane, status line, or empty space to open context menus[2]. The wheel bindings are worth understanding: `WheelUpPane` enters copy mode and scrolls up unless the pane is in the alternate screen (a full-screen app like `vim` or `less`), in which case the wheel is passed through to the app[2].

From the shell: `tmux join-pane -s :2` pulls window 2 in as a pane here, `tmux setw synchronize-panes` toggles typing into every pane at once, `tmux capture-pane -p -S -1000` dumps the last 1000 lines of a pane, and `tmux pipe-pane -o 'cat >> ~/pane.log'` logs pane output to a file[9][10]. The marked pane (`C-b m`) is the silent workhorse of pane plumbing: once marked, `join-pane`, `move-pane`, `swap-pane` and `swap-window` use it as their default source, so you can mark a pane in one window and pull it into another without naming targets[1].

Copy mode: the emacs table

Copy mode freezes a pane so you can scroll its history, select text, and copy it to a paste buffer. Enter it with `C-b [` (or `C-b PgUp` to jump straight up a page); leave with `q` (or Escape in emacs). The top-right corner shows your position and the total history lines, capped by `history-limit` (default **2000 lines** per pane)[1][5].

With `mode-keys` at its default of **emacs**, the active table is `copy-mode`:

KEY	ACTION
`C-Space`	Begin selection
`C-a` / `C-e`	Start / end of line
`C-f` / `C-b`	Right / left one character
`C-n` / `C-p`	Down / up one line
`C-v` / `M-v`	Page down / page up
`M-w`	Copy selection and exit (`copy-pipe-and-cancel`)
`C-w`	Copy and exit (same action)
`C-k`	Copy from cursor to end of line and exit
`C-r` / `C-s`	Incremental search backward / forward
`M-<` / `M->`	Jump to top / bottom of history
`M-f` / `M-b`	Next word end / previous word
`C-Up` / `C-Down`	Scroll up / down one line
`M-Up` / `M-Down`	Scroll half a page
`n` / `N`	Repeat search / repeat reversed
`L`	Toggle copy-mode line numbers (3.7)
`R`	Toggle rectangle (block) selection
`P`	Toggle the position indicator
`X`	Mark the current line (`set-mark`); `M-x` jumps to the mark
`C-l`	Re-centre the cursor (top/centre/bottom, 3.7)
`M-l`	Move cursor to horizontal centre (3.7)

KEY	ACTION
`f` `F` `t` `T` `g`	Jump forward/backward, jump-to, goto line
`Escape`, `q`	Cancel / exit copy mode

Every default emacs copy-mode binding (from key-bindings.c [2])

KEY	ACTION
`C-Space`	begin-selection
`C-a` / `C-e`	start-of-line / end-of-line
`C-b` / `C-f`	cursor-left / cursor-right
`C-n` / `C-p`	cursor-down / cursor-up
`C-c`, `Escape`, `C-[`, `q`	cancel
`C-g`	clear-selection
`C-k`	copy-pipe-end-of-line-and-cancel
`C-l`	recentre-top-bottom
`M-l`	cursor-centre-horizontal
`C-r` / `C-s`	search-backward / forward-incremental
`C-v` / `M-v`	page-down / page-up
`C-w` / `M-w`	copy-pipe-and-cancel
`Space`	page-down
`,` / `;`	jump-reverse / jump-again
`F` / `f`	jump-backward / jump-forward
`T` / `t`	jump-to-backward / jump-to-forward
`g`	goto-line
`L`	line-numbers-toggle
`N`	search-reverse
`n`	search-again
`P`	toggle-position
`R`	rectangle-toggle
`X`	set-mark
`r`	refresh-now
`Home` / `End`	start-of-line / end-of-line

KEY	ACTION
`M-1` … `M-9`	repeat count
`M-<` / `M->`	history-top / history-bottom
`M-R`	top-line
`M-r`	middle-line
`M-b` / `M-f`	previous-word / next-word-end
`C-M-b` / `C-M-f`	previous / next matching bracket
`M-{` / `M-}`	previous / next paragraph
`M-m`	back-to-indentation
`M-x`	jump-to-mark
`M-Up` / `M-Down`	halfpage-up / halfpage-down
`C-Up` / `C-Down`	scroll-up / scroll-down
`M-C-Up` / `M-C-Down`	previous-prompt / next-prompt (OSC 133)
Mouse	drag selects, drag-end copies, wheel scrolls 5 lines, double/triple click copy word/line

The incremental searches (`C-r` / `C-s`) are designed to emulate emacs: first press starts a search, an empty repeat reuses the last term[1]. Since 3.5, copy mode shows OSC 8 hyperlinks and exposes the link under the cursor as `copy_cursor_hyperlink`; the jump-to-mark machinery (`X` to mark a line, `M-x` to jump back) dates to 3.2[3].

Copy mode: the vi table

If `EDITOR` or `VISUAL` contains “vi” (or you set `set -g mode-keys vi`), copy mode uses the `copy-mode-vi` table and feels like vim. This is the table most people actually learn:

KEY	ACTION
`h j k l` / arrows	Move cursor
`w e b` / `W E B`	Word navigation (space-separated variants)
`0` / `$` / `^`	Start of line / end / first non-blank
`Space`	Begin selection
`Enter`	Copy selection and exit copy mode
`v` / `C-v`	Toggle rectangle selection
`V`	Select the whole line
`/` / `?`	Search forward / backward
`n` / `N`	Repeat search / reversed

KEY	ACTION
`g` / `G`	Top / bottom of history
`H` `M` `L`	Top / middle / bottom line
`f` `F` `t` `T`	Jump within the line; `;` and `,` repeat
`%`	Matching bracket
`{` / `}`	Previous / next paragraph
`*` / `#`	Search forward / backward for the word under the cursor
`C-b` / `C-f`	Page up / down
`C-u` / `C-d`	Half page up / down
`C-e` / `C-y`	Scroll down / up one line
`o`	Jump to the other end of the selection
`z`	Scroll the cursor line to the middle
`q`	Exit copy mode

Every default vi copy-mode binding (from key-bindings.c [2])

KEY	ACTION
`#` / `*`	search-backward / forward for word under cursor
`C-c` , `q`	cancel
`C-d` / `C-u`	halfpage-down / halfpage-up
`C-e` / `C-y`	scroll-down / scroll-up
`C-b` / `C-f`	page-up / page-down
`C-h`	cursor-left
`C-j` , `Enter`	copy-pipe-and-cancel
`C-v` / `v`	rectangle-toggle
`Escape` , `C-[`	clear-selection
`Space`	begin-selection
`$` / `0`	end-of-line / start-of-line
`,` / `;`	jump-reverse / jump-again
`/` / `?`	search-forward / backward
`:`	goto-line
`1` – `9`	repeat count

KEY	ACTION
`A`	append-selection-and-cancel
`B` / `E` / `W`	previous-space / next-space-end / next-space
`D`	copy-pipe-end-of-line-and-cancel
`F` / `f`	jump-backward / jump-forward
`T` / `t`	jump-to-backward / jump-to-forward
`G` / `g`	history-bottom / history-top
`H` / `M` / `L`	top-line / middle-line / bottom-line
`J` / `K`	scroll-down / scroll-up
`N` / `n`	search-reverse / search-again
`P`	toggle-position
`V`	select-line
`X`	set-mark
`^`	back-to-indentation
`b` / `e` / `w`	previous-word / next-word-end / next-word
`h` `j` `k` `l`	cursor-left / down / up / right
`z`	scroll-middle
`o`	other-end
`r`	refresh-now
`{` / `}`	previous / next paragraph
`%`	next-matching-bracket
`BSpace`	cursor-left
`Home` / `End`	start-of-line / end-of-line
`NPage` / `PPage`	page-down / page-up
`M-x`	jump-to-mark
`C-Up` / `C-Down`	scroll-up / scroll-down
Mouse	same drag/scroll/copy bindings as emacs

Two practical notes. First, in the default vi table `v` **toggles rectangle selection — it does not start one**; `Space` starts selection, `V` selects a line, `Enter` copies and exits[9]. Second, since 3.4, if the shell emits OSC 133 prompt markers, copy mode can jump between shell prompts (`previous-prompt` / `next-prompt`, bound to `M-C-Up`/`M-C-Down` in emacs; the vi table leaves them unbound — the canonical recipe from the 3.4 changelog is `bind -T copy-mode-vi '[' send -X previous-prompt` and `bind -T copy-mode-vi ']' send -X next-prompt`)[1][7]. Since 3.7, copy

mode can also show line numbers (`L`, or the `copy-mode-line-numbers` option: `off`, `default`, `absolute`, `relative`, `hybrid`) and re-centre the cursor (`C-l` in emacs)[3][7].

One difference between the tables that trips people up: in emacs mode the arrow keys move the cursor and `M-Up`/`M-Down` scroll half pages, while in vi mode the arrow keys also move and `C-u`/`C-d` scroll half pages — but the *scroll* commands (`C-e`, `C-y`, `J`, `K`) keep the cursor fixed relative to the text, whereas cursor movement commands move through the text. Since 3.7, vi scrolling keeps the cursor at the same position relative to the text, closer to real vim behaviour[3]. And the `copy-pipe-and-cancel` behind `Enter`/`C-j` is why `y`-style copy works: the default copies the selection into a paste buffer and exits; if you want the selection piped to a command (say, straight into your system clipboard), that is what the `copy-pipe` variants and the `copy-command` server option are for[1][11].

Paste buffers, the command prompt, and diagnostics

Everything you copy lands in a tmux **paste buffer** (named `buffer0001`, `buffer0002`, …), and buffers persist across panes and windows. tmux keeps up to **50** automatic buffers (`buffer-limit`, a server option); when you pass 50, the oldest is deleted[11]. Explicitly named buffers are exempt from the limit[1].

KEY	ACTION	COMMAND BEHIND IT
`C-b ]`	Paste the most recent buffer	`paste-buffer -p`
`C-b #`	List all paste buffers	`list-buffers`
`C-b =`	Pick a buffer to paste, interactively	`choose-buffer -Z`
`C-b -`	Delete the most recent buffer	`delete-buffer`
`C-b :`	Open the tmux command prompt	`command-prompt`
`C-b ?`	List all key bindings	`list-keys -N`
`C-b /`	Describe what a key does (3.1)	`list-keys -1N`
`C-b C`	Browse and edit options interactively (3.2)	`customize-mode -Z`
`C-b i`	Show info about the current window	`display-message`
`C-b t`	Show a big clock (any key dismisses)	`clock-mode`
`C-b ~`	Show the server message log	`show-messages`
`C-b r`	Force a redraw of the client	`refresh-client`
`C-b C-z`	Suspend the tmux client (bring back with `fg`)	`suspend-client`

The `:` command prompt is where you type tmux commands directly — `C-b : set -g mouse on`, `C-b : source-file ~/.tmux.conf` to reload your config. It honours `status-keys` (emacs unless `EDITOR` contains “vi”)[1]:

FUNCTION	VI	EMACS
Execute	`Enter`	`Enter`
Cancel	`q`	`Escape`
Move cursor left / right	`h` / `l`	`←` / `→`
Start / end of line	`0` / `$`	`C-a` / `C-e`
Word left / right	`b` / `w`	`M-b` / `M-f`
Delete to start / end	`d` / `D`	`C-u` / `C-k`
Insert top paste buffer	`p`	`C-y`
Tab-complete	`Tab`	`Tab`

Buffer commands from the shell: `tmux lsb` (list), `tmux showb` (print newest), `tmux saveb file` and `tmux loadb file` (save/load to disk)[9]. `C-b =` is the interactive buffer chooser; inside it, `p` pastes the selected buffer, `d` deletes, `e` opens it in your `$EDITOR` (default `vi`[11]), and `Enter` pastes and closes[1]. `C-b C` (customize mode, added in 3.2[3]) is the interactive options browser — search, filter and edit any option or key binding from a list, which is the fastest way to discover what an option does without reading the man page. `C-b ~` shows the server message log (since 3.2 the log is per-server, not per-client, so it survives detaches)[3]. And `C-b C-z` is the one that makes tmux “disappear”: it suspends the client, dropping you back to the shell that started tmux; `fg` brings it back[9].

What changed recently, and keeping this accurate

The prefix table is remarkably stable, but a few releases added or changed default bindings. Here is the honest version history — several bindings people assume are brand new are actually old:

2017 — **tmux 2.7**
C-b E (select-layout -E) added

13 Feb 2024 — **tmux 3.4**
SIXEL images, styled menus, OSC 133 prompt marking, double/triple-click copy

27 Sep 2024 — **tmux 3.5**
escape-time cut to 10 ms, prefix-timeout, mirrored main layouts

26 Nov 2025 — **tmux 3.6**
pane scrollbars, dark/light theme support, copy-mode selection styles

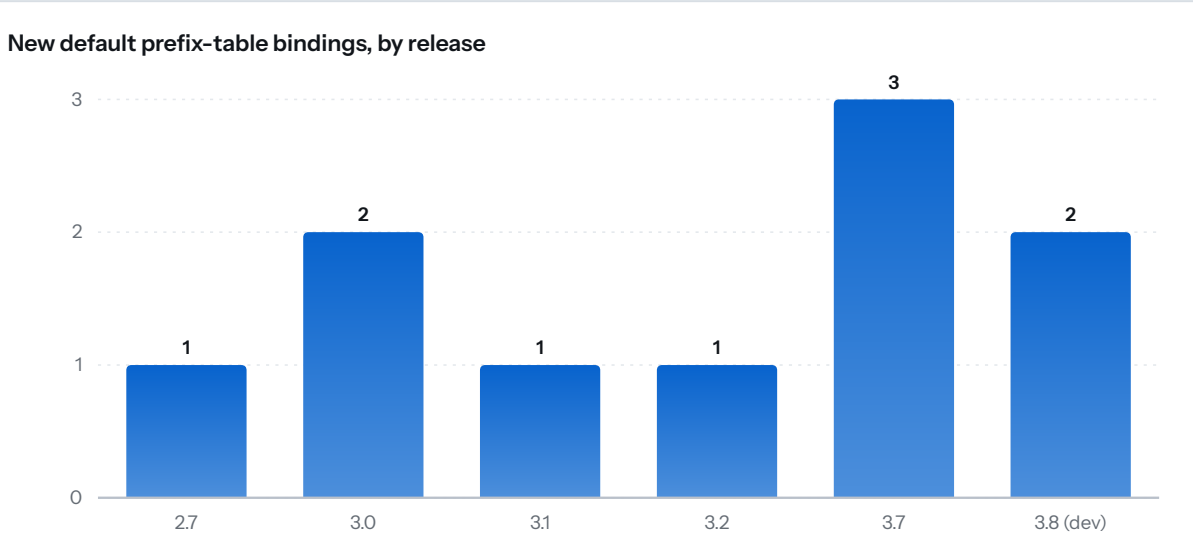
26 Jun 2026 — **tmux 3.7**
Floating panes (C-b *), switch-mode (C-b Tab / BTab), copy-mode line numbers

1 Jul 2026 — **tmux 3.7b**
Current stable release

In development

tmux 3.8

C-b @ float toggle, C-b g move table, mouse on by default



Counts from the CHANGES file [3]: 2.7 added E; 3.0 added < and > menus; 3.1 added / describe-key; 3.2 added C customize-mode; 3.7 added *, Tab and BTab; 3.8 adds @ and the g move table.

Version facts worth remembering:

- `C-b <` **and** `C-b >` (window and pane context menus) date to **3.0** — "Default menus are bound to MouseDown3 on the status line; MouseDown3 or M-MouseDown3 on panes; MouseDown3 in tree, client and buffer modes; and C-b < and >" [3]. They are not new.
- `C-b E` (spread panes evenly) dates to **2.7**: "Add select-layout -E to spread panes out evenly (bound to E key)" [3].
- `C-b /` (describe a key) dates to **3.1**, added with key notes: "Add support for adding a note to a key binding … add a default binding (/) to show the note for the next key pressed" [3].
- `C-b C` (customize mode, the interactive options browser) dates to **3.2**: "Add a customize mode where options may be browsed and changed … Bound to C-b C by default" [3].
- **3.7 (26 June 2026)** added floating panes (`C-b *`), the Tab/BTab fuzzy switcher, and copy-mode line numbers [3] [7].
- **3.7b (1 July 2026)** is the current stable release [6] [8].
- **3.8 is unreleased** as of 11 August 2026 (master branch). Its confirmed new defaults: `C-b @` to float/tile the current pane, the `C-b g` floating-pane move table, and `mouse` defaulting to on [2] [3]. If your distro ships 3.7b, those keys are not there yet.

KEY FINDING

When a cheat sheet tells you something is "new", check it against `tmux lsk -N` on your own install. The `mouse` option defaults to **off** through 3.7 and **on** in 3.8 — if you upgrade to 3.8 you will suddenly get click-to-select, drag-resize and wheel-scroll behaviour you did not configure. `set -g mouse off` restores the old feel.

> **CAUTION**
> A running tmux server keeps its old behaviour after an upgrade — bindings, options and all. The 3.7 release notes repeat the standing advice: “fully restart tmux (kill-server) after upgrading so every session runs the new binary”[7][8].

To verify anything in this document against your exact build: `tmux -V` for the version, `tmux lsk -N` for every binding with its note, and `tmux lsk -T prefix` / `-T copy-mode-vi` for a single table[2][5]. The man page (`man 1 tmux`, section “DEFAULT KEY BINDINGS”) is the authoritative text; the OpenBSD mirror tracks the current release[1][4]. The default-bindings source (`key-bindings.c` in the tmux repo) is the ground truth behind both[2], and the option defaults (buffer-limit 50, escape-time 10 ms, exit-empty on) come from `options-table.c` in the same tree[11].

Bottom line

The default prefix is `C-b`: `C-b c` new window, `C-b %` / `C-b "` splits, `C-b d` detach, `C-b [` copy mode, `C-b ]` paste, `C-b ?` to see everything. If you only memorise a handful, those six plus `C-b ,` (rename window), `C-b z` (zoom), and `C-b 0`–`9` (jump windows) cover 90% of daily use. The full table is stable across versions — the only real churn is 3.7’s floating panes and switch-mode, and the unreleased 3.8’s `@`, `g` and mouse-on-default — so this reference stays accurate against 3.7b, and `tmux lsk -N` will always show you the ground truth for whatever you actually run. Want a different prefix? `set -g prefix C-a` and `unbind C-b` in `~/.tmux.conf`, then `C-b :` `source-file ~/.tmux.conf`.

SOURCES

1 tmux(1) man page (OpenBSD, current release)
https://man.openbsd.org/tmux

2 tmux key-bindings.c — default key bindings source
https://raw.githubusercontent.com/tmux/tmux/master/key-bindings.c

3 tmux CHANGES — full changelog
https://raw.githubusercontent.com/tmux/tmux/master/CHANGES

4 tmux(1) man page (man7.org)
https://man7.org/linux/man-pages/man1/tmux.1.html

5 Getting Started — tmux/tmux Wiki (official)
https://github.com/tmux/tmux/wiki/Getting-Started

6 Releases — tmux/tmux (GitHub)
https://github.com/tmux/tmux/releases

7 What’s New in tmux 3.7 — tmux.info
https://www.tmux.info/docs/whats-new

8 Release Notes — tmux.info
https://www.tmux.info/releases

9 tmux cheat sheet — tmuxcheatsheet.dev
https://tmuxcheatsheet.dev/

10 tmux Cheat Sheet & Shortcuts — tmux.app
https://tmux.app/cheat-sheet/

11 tmux options-table.c — option defaults source
https://raw.githubusercontent.com/tmux/tmux/master/options-table.c

Researched and written with deepseek/deepseek-v4-flash-0731, from the sources listed above. Figures are as reported by those sources. Exported November 29, 2026 from https://intel.innerpulse.net/doc/tmux-default-prefix-key-bindings-a-quick-reference-for-ctrl-b, where the current version lives.