tmuxinator tmux session templates

START HERE

`tmuxinator` **new** *<project>*
Create a project file

`tmuxinator` **doctor**
Check your setup

`tmuxinator` **start** *<project>*
Launch its session

alias mux=tmuxinator
Shorter command

`tmuxinator` **edit** *<project>*
Change the layout

`tmuxinator` **list**
See all projects

`tmuxinator` **stop** *<project>*
Kill its session

`tmuxinator` **new --local**
Config in this repo

CONCEPTS

A project is a YAML file that describes one tmux session: windows, panes and the commands each runs.

Project file name, root, hooks
Window layout, root
Pane command

tmuxinator only builds the session; once it runs, you work in plain tmux.

YAML indentation matters: a window name and its settings are nested one level below the window list entry.

PROJECTS

`tmuxinator` **new** *<project>*
Create a project and open it in $EDITOR

`tmuxinator` **edit** *<project>*
Edit an existing project file

`tmuxinator` **list**
List configured projects

`tmuxinator` **open** *<project>*
Open a project, creating it if missing

`tmuxinator` **copy** *<old>* *<new>*
Duplicate a project

`tmuxinator` **delete** *<project>*
Remove a project file

PROJECTS (cont.)

`tmuxinator` **new** *<project>* *<session>*
Build a project from a running tmux session
Records windows, layouts, panes and root of the live session.

`tmuxinator` **implode**
Remove all configs, aliases and scripts
Destructive: deletes every project you have.

START AND STOP

`tmuxinator` **start** *<project>*
Create the session and attach

`tmuxinator` **start**
Start from ./.tmuxinator.yml

`tmuxinator` **stop** *<project>*
Stop a project's session

`tmuxinator` **stop-all**
Stop every active project

`tmuxinator` **start** *<project>* **-n** *<name>*
Use a custom session name

`tmuxinator` **start** *<project>* **-p** *<file>*
Use a specific config file
Overrides the project name and any local file.

`tmuxinator` **start** *<project>* **--append**
Add windows to the current session

`tmuxinator` **start** *<project>* *<arg>*
Pass positional args to the template

START AND STOP (cont.)

`tmuxinator` **start** *<project>* *<key>=<value>*
Pass named settings to the template

PROJECT FILE: TOP LEVEL

name: *<name>*
Session name (no dots)
Dots are not allowed: tmux uses them to separate windows and panes.

root: ~/code/app
Directory every window starts in

windows:
List of window definitions

startup_window: *<name>*
Window selected on start

startup_pane: *<index>*
Pane selected in that window

pre_window: *<cmd>*
Run before every pane's commands

attach: false
Create the session without attaching

tmux_options: -f *<file>*
Extra options passed to tmux

WINDOWS AND PANES

- editor: vim
Window with a name and one command

WINDOWS AND PANES (cont.)

layout: main-vertical
Pane layout for the window
Also even-horizontal, even-vertical, main-horizontal, tiled, or a custom layout string.

panes:
List of panes in the window

- logs: tail -f log/dev.log
Named pane running a command

root: *<dir>*
Window-specific root directory

focused_pane: *<index or name>*
Pane that gets focus on start

synchronize: before
Type into all panes at once
before: on from the start; after: on once the pane commands have run.

- ssh me@host / cd /var/log
Several commands via send-keys
List the commands under a key; works over SSH.

HOOKS AND TEMPLATES

on_project_start: *<cmd>*
Run on every start

on_project_first_start: *<cmd>*
Run only when the session is created

HOOKS AND TEMPLATES (cont.)

on_project_exit: *<cmd>*
Run when you detach

on_project_stop: *<cmd>*
Run on tmuxinator stop

<%= ENV["<VAR>"] %>
Insert an environment variable

<%= @args[0] %>
Insert a positional argument

<%= @settings["<key>"] %>
Insert a key=value setting

SETUP AND DEBUG

`tmuxinator` **doctor**
Check tmux, editor and shell setup

`tmuxinator` **debug** *<project>*
Print the shell commands it would run

`tmuxinator` **version**
Show the installed version

`tmuxinator` **help** *<command>*
Help for one command

gem install tmuxinator
Install via RubyGems

export EDITOR=*<editor>*
Editor used by new, open, edit

export TMUXINATOR_CONFIG=*<dir>*
Config dir, checked first
Order: $TMUXINATOR_CONFIG, $XDG_CONFIG_HOME/tmuxinator, ~/.tmuxinator.

SETUP AND DEBUG (cont.)

A ./.tmuxinator.yml in the current directory is used when no named project matches.

If pane commands get lost or garbled, move them into a script and source it from the pane.

DISABLE_AUTO_TITLE=true in your shell RC fixes wrong window names.

RECIPES

Dev project

```
name: app
root: ~/code/app
startup_window: editor
windows:
  - editor: vim
  - server: bin/rails server
  - logs: tail -f log/
development.log
```

Editor with side panes

```
windows:
  - main:
      layout: main-vertical
      panes:
        - vim
        - git status
        - guard
```

Per-repo setup

```
cd ~/code/app
tmuxinator new --local app
tmuxinator start
```

Parametrised start

```
# in the YAML: root: <%=
@settings["dir"] %>
tmuxinator start work dir=~/
code/api
```