

Configure the tiling window manager yabai for macos

Ronald (Ryy) Glenn Thomas

2024-02-02

Table of contents

1	Introduction	1
2	skhd	5

1 Introduction

Yabai is a tiling window manager for macOS.

From the wiki [Home](#) · [koekeishiya/yabai Wiki](#) · [GitHub](#)

“It automatically modifies your window layout using a binary space partitioning algorithm interface allows you to control and query windows, spaces and displays to enable powerful integration with tools like **skhd** to allow you to work more efficiently with macOS.”

Requirements: (from the WIKI)

- In the Desktop & Dock tab, inside the Mission Control pane, the setting “Displays have separate Spaces” must be enabled.



Figure 1: quarto

- System Settings (macOS 13.x, 14.x) In the Desktop & Dock tab, inside the Mission Control pane, the setting “Automatically rearrange Spaces based on most recent use” should be disabled for commands that rely on the ordering of spaces to work reliably.
- System Settings (macOS 14.x) In the Desktop & Dock tab, inside the Desktop & Stage Manager pane, the setting “Show Items On Desktop” should be enabled for display and space focus commands to work reliably in multi-display configurations.
- System Settings (macOS 14.x) In the Desktop & Dock tab, inside the Desktop & Stage Manager pane, the setting “Click wallpaper to reveal Desktop” should be set to “Only in Stage Manager” for display and space focus commands to work reliably.

```
brew install koekeishiya/formulae/yabai
# start yabai
yabai --start-service
```

Sample configuration files can be found in the `examples` directory. Refer to the `documentation` or the `wiki` for further information.

Keyboard shortcuts can be defined with `skhd` or any other suitable software you may prefer. `## yabairc`

```
#!/usr/bin/env bash
sudo yabai --load-sa
yabai -m signal --add event=dock_did_restart action="sudo yabai --load-sa"
set -x

# ===== Variables =====

declare -A gaps
declare -A color

gaps["top"]="4"
gaps["bottom"]="24"
```

```

gaps["left"]="4"
gaps["right"]="4"
gaps["inner"]="4"

color["focused"]="0xE0808080"
color["normal"]="0x00010101"
color["preselect"]="0xE02d74da"

# Uncomment to refresh ubersicht widget on workspace change
# Make sure to replace WIDGET NAME for the name of the ubersicht widget
#ubersicht_spaces_refresh_command="osascript -e 'tell application id \"tracesOf.Uebersicht\" t

# ===== Loading Scripting Additions =====

# See: https://github.com/koekeishiya/yabai/wiki/Installing-yabai-\(latest-release\)#macos-big-s
sudo yabai --load-sa
yabai -m signal --add event=dock_did_restart action="sudo yabai --load-sa"

# ===== Tiling setting =====

yabai -m config layout                bsp

yabai -m config top_padding           "${gaps["top"]}"
yabai -m config bottom_padding       "${gaps["bottom"]}"
yabai -m config left_padding          "${gaps["left"]}"
yabai -m config right_padding         "${gaps["right"]}"
yabai -m config window_gap           "${gaps["inner"]}"

yabai -m config mouse_follows_focus   off
yabai -m config focus_follows_mouse   off

yabai -m config window_topmost        off
yabai -m config window_opacity        off
yabai -m config window_shadow         float

yabai -m config window_border         on
yabai -m config window_border_width   2
yabai -m config active_window_border_color "${color["focused"]}"
yabai -m config normal_window_border_color "${color["normal"]}"
yabai -m config insert_feedback_color  "${color["preselect"]}"

```

```

yabai -m config active_window_opacity      1.0
yabai -m config normal_window_opacity      0.90
yabai -m config split_ratio                0.50

yabai -m config auto_balance                on

yabai -m config mouse_modifier              fn
yabai -m config mouse_action1              move
yabai -m config mouse_action2              resize

# ===== Rules =====

yabai -m rule --add label="Finder" app="^Finder$" title="(Co(py|nnect)|Move|Info|Pref)" manage=off
yabai -m rule --add label="Safari" app="^Safari$" title="^(General|(Tab|Password|Website|Extension))" manage=off
yabai -m rule --add label="macfeh" app="^macfeh$" manage=off
yabai -m rule --add label="System Preferences" app="^System Preferences$" title=".*" manage=off
yabai -m rule --add label="App Store" app="^App Store$" manage=off
yabai -m rule --add label="Activity Monitor" app="^Activity Monitor$" manage=off
yabai -m rule --add label="KeePassXC" app="^KeePassXC$" manage=off
yabai -m rule --add label="Calculator" app="^Calculator$" manage=off
yabai -m rule --add label="Dictionary" app="^Dictionary$" manage=off
yabai -m rule --add label="mpv" app="^mpv$" manage=off
yabai -m rule --add label="Software Update" title="Software Update" manage=off
yabai -m rule --add label="About This Mac" app="System Information" title="About This Mac" manage=off

# ===== Signals =====

yabai -m signal --add event=application_front_switched action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=display_changed action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=space_changed action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=window_created action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=window_destroyed action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=window_focused action="${ubersicht_spaces_refresh_command}"
yabai -m signal --add event=window_title_changed action="${ubersicht_spaces_refresh_command}"

set +x
printf "yabai: configuration loaded...\n"

```

2 skhd

```
# opens iTerm2
alt - return : "${HOME}"/bin/open_iterm.sh

# Show system statistics
fn + lalt - 1 : "${HOME}"/.config/yabai/scripts/show_cpu.sh
fn + lalt - 2 : "${HOME}"/.config/yabai/scripts/show_mem.sh
fn + lalt - 3 : "${HOME}"/.config/yabai/scripts/show_bat.sh
fn + lalt - 4 : "${HOME}"/.config/yabai/scripts/show_disk.sh
fn + lalt - 5 : "${HOME}"/.config/yabai/scripts/show_song.sh

# Navigation
alt - h : yabai -m window --focus west
alt - j : yabai -m window --focus south
alt - k : yabai -m window --focus north
alt - l : yabai -m window --focus east
alt - 1 : yabai -m space --focus 1
alt - 2 : yabai -m space --focus 2
alt - 3 : yabai -m space --focus 3
alt - 4 : yabai -m space --focus 4
alt - 5 : yabai -m space --focus 5
alt - 6 : yabai -m space --focus 6
alt - 7 : yabai -m space --focus 7
alt - 8 : yabai -m space --focus 8

# Moving windows
shift + alt - h : yabai -m window --warp west
shift + alt - j : yabai -m window --warp south
shift + alt - k : yabai -m window --warp north
shift + alt - l : yabai -m window --warp east

# Move focus container to workspace
shift + alt - m : yabai -m window --space last; yabai -m space --focus last
shift + alt - p : yabai -m window --space prev; yabai -m space --focus prev
shift + alt - n : yabai -m window --space next; yabai -m space --focus next
shift + alt - 1 : yabai -m window --space 1; yabai -m space --focus 1
shift + alt - 2 : yabai -m window --space 2; yabai -m space --focus 2
shift + alt - 3 : yabai -m window --space 3; yabai -m space --focus 3
shift + alt - 4 : yabai -m window --space 4; yabai -m space --focus 4
```

```

# Resize windows
lctrl + alt - h : yabai -m window --resize left:-50:0; \
                  yabai -m window --resize right:-50:0
lctrl + alt - j : yabai -m window --resize bottom:0:50; \
                  yabai -m window --resize top:0:50
lctrl + alt - k : yabai -m window --resize top:0:-50; \
                  yabai -m window --resize bottom:0:-50
lctrl + alt - l : yabai -m window --resize right:50:0; \
                  yabai -m window --resize left:50:0

# Equalize size of windows
lctrl + alt - e : yabai -m space --balance

# Enable / Disable gaps in current workspace
lctrl + alt - g : yabai -m space --toggle padding; yabai -m space --toggle gap

# Rotate windows clockwise and anticlockwise
alt - r          : yabai -m space --rotate 270
shift + alt - r : yabai -m space --rotate 90

# Rotate on X and Y Axis
shift + alt - x : yabai -m space --mirror x-axis
shift + alt - y : yabai -m space --mirror y-axis

# Set insertion point for focused container
shift + lctrl + alt - h : yabai -m window --insert west
shift + lctrl + alt - j : yabai -m window --insert south
shift + lctrl + alt - k : yabai -m window --insert north
shift + lctrl + alt - l : yabai -m window --insert east

# Float / Unfloat window
shift + alt - space : \
    yabai -m window --toggle float; \
    yabai -m window --toggle border

# Restart Yabai
shift + lctrl + alt - r : \
    /usr/bin/env osascript <<< \
        "display notification \"Restarting Yabai\" with title \"Yabai\"; \
    launchctl kickstart -k "gui/${UID}/homebrew.mxcl.yabai"

```

```

# Make window native fullscreen
alt - f      : yabai -m window --toggle zoom-fullscreen
shift + alt - f : yabai -m window --toggle native-fullscreen

# Make floating window fill screen
shift + alt - up      : chunkc tiling::window --grid-layout 1:1:0:0:1:1

# make floating window fill left-half of screen
shift + alt - left    : chunkc tiling::window --grid-layout 1:2:0:0:1:1

# make floating window fill right-half of screen
shift + alt - right   : chunkc tiling::window --grid-layout 1:2:1:0:1:1

# create desktop, move window and follow focus
shift + cmd - n : chunkc tiling::desktop --create;\
                  id=$(chunkc tiling::query --desktops-for-monitor $(chunkc tiling::query --monitors-for-window $(chunkc tiling::query --window-id)))\
                  chunkc tiling::window --send-to-desktop $(echo ${id##* });\
                  chunkc tiling::desktop --focus $(echo ${id##* })

# create desktop and follow focus
cmd + alt - n : chunkc tiling::desktop --create;\
                id=$(chunkc tiling::query --desktops-for-monitor $(chunkc tiling::query --monitors-for-window $(chunkc tiling::query --window-id)))\
                chunkc tiling::desktop --focus $(echo ${id##* })

# destroy desktop
cmd + alt - w : chunkc tiling::desktop --annihilate

# make a workspace float
alt - 2 : yabai -m config --space 2 layout float
alt - 3 : yabai -m config --space 3 layout float

```