

NAME

bspwm – Binary space partitioning window manager

SYNOPSIS

bspwm [-h|-v|-c *CONFIG_PATH*]

bspc --print-socket-path

bspc *DOMAIN* [*SELECTOR*] *COMMANDS*

bspc *COMMAND* [*OPTIONS*] [*ARGUMENTS*]

DESCRIPTION

bspwm is a tiling window manager that represents windows as the leaves of a full binary tree.

It is controlled and configured via **bspc**.

OPTIONS

-h

Print the synopsis and exit.

-v

Print the version and exit.

-c *CONFIG_PATH*

Use the given configuration file.

--print-socket-path

Print the **bspwm** socket path and exit.

COMMON DEFINITIONS

DIR := north | west | south | east

CYCLE_DIR := next | prev

SELECTORS

Selectors are used to select a target node, desktop, or monitor. A selector can either describe the target relatively or name it globally.

Selectors consist of an optional reference, a descriptor and any number of non-conflicting modifiers as follows:

[REFERENCE#]DESCRIPTOR(.MODIFIER)*

The relative targets are computed in relation to the given reference (the default reference value is **focused**).

An exclamation mark can be prepended to any modifier in order to reverse its meaning.

The following characters cannot be used in monitor or desktop names: #, :, ..

The special selector %<name> can be used to select a monitor or a desktop with an invalid name.

Node

Select a node.

NODE_SEL := [NODE_SEL#](DIR|CYCLE_DIR|PATH|any|first_ancestor|last|newest|older|newer|focused|pointed|biggest|smallest|<node_id>)[.(!)]focused[.(!)]active[.(!)]automatic[.(!)]local[.(!)]leaf[.(!)]window[.(!)]STATE[.(!)]FLAG[.(!)]LAYER[.(!)]SPLIT_TYPE[.(!)]same_class[.(!)]descendant_of[.(!)]ancestor_of

STATE := tiled|pseudo_tiled|floating|fullscreen

FLAG := hidden|sticky|private|locked|marked|urgent

LAYER := below|normal|above

SPLIT_TYPE := horizontal|vertical

PATH := @[DESKTOP_SEL:][[/]JUMP](/JUMP)*

JUMP := first|1|second|2|brother|parent|DIR

Descriptors

DIR

Selects the window in the given (spacial) direction relative to the reference node.

CYCLE_DIR

Selects the node in the given (cyclic) direction relative to the reference node within a depth-first in-order traversal of the tree.

PATH

Selects the node at the given path.

any

Selects the first node that matches the given selectors.

first_ancestor

Selects the first ancestor of the reference node that matches the given selectors.

last

Selects the previously focused node relative to the reference node.

newest

Selects the newest node in the history of the focused node.

older

Selects the node older than the reference node in the history.

newer

Selects the node newer than the reference node in the history.

focused

Selects the currently focused node.

pointed

Selects the leaf under the pointer.

biggest

Selects the biggest leaf.

smallest

Selects the smallest leaf.

<node_id>

Selects the node with the given ID.

Path Jumps

The initial node is the focused node (or the root if the path starts with /) of the reference desktop (or the selected desktop if the path has a *DESKTOP_SEL* prefix).

1|first

Jumps to the first child.

2|second
Jumps to the second child.

brother
Jumps to the brother node.

parent
Jumps to the parent node.

DIR
Jumps to the node holding the edge in the given direction.

Modifiers

[!]focused
Only consider the focused node.

[!]active
Only consider nodes that are the focused node of their desktop.

[!]automatic
Only consider nodes in automatic insertion mode. See also **--presel-dir** under **Node** in the **DOMAINS** section below.

[!]local
Only consider nodes in the reference desktop.

[!]leaf
Only consider leaf nodes.

[!]window
Only consider nodes that hold a window.

[!](tiled|pseudo_tiled|floating|fullscreen)
Only consider windows in the given state.

[!]same_class
Only consider windows that have the same class as the reference window.

[!]descendant_of
Only consider nodes that are descendants of the reference node.

[!]ancestor_of
Only consider nodes that are ancestors of the reference node.

[!](hidden|sticky|private|locked|marked|urgent)
Only consider windows that have the given flag set.

[!](below|normal|above)
Only consider windows in the given layer.

[!](horizontal|vertical)
Only consider nodes with the given split type.

Desktop

Select a desktop.

```
DESKTOP_SEL := [DESKTOP_SEL#](CYCLE_DIR|any|last|newest|older|newer|
[MONITOR_SEL:](focused|<n>)|
<desktop_id>|<desktop_name>)[.(!)focused][.(!)active]
[.(!)occupied][.(!)urgent][.(!)local]
[.(!)LAYOUT][.(!)user_LAYOUT]
```

LAYOUT := tiled|monocle

Descriptors

CYCLE_DIR

Selects the desktop in the given direction relative to the reference desktop.

any

Selects the first desktop that matches the given selectors.

last

Selects the previously focused desktop relative to the reference desktop.

newest

Selects the newest desktop in the history of the focused desktops.

older

Selects the desktop older than the reference desktop in the history.

newer

Selects the desktop newer than the reference desktop in the history.

focused

Selects the currently focused desktop.

^<n>

Selects the nth desktop. If **MONITOR_SEL** is given, selects the nth desktop on the selected monitor.

<desktop_id>

Selects the desktop with the given ID.

<desktop_name>

Selects the desktop with the given name.

Modifiers

[!]focused

Only consider the focused desktop.

[!]active

Only consider desktops that are the focused desktop of their monitor.

[!]occupied

Only consider occupied desktops.

[!]urgent

Only consider urgent desktops.

[!]local

Only consider desktops inside the reference monitor.

[!](tiled|monocle)

Only consider desktops with the given layout.

[!](user_tiled|user_monocle)

Only consider desktops which have the given layout as userLayout.

Monitor

Select a monitor.

```
MONITOR_SEL := [MONITOR_SEL#](DIR|CYCLE_DIR|any|last|newest|older|newer|
    focused|pointed|primary|^<n>|
    <monitor_id>|<monitor_name>)[.![!]focused][.![!]occupied]
```

Descriptors*DIR*

Selects the monitor in the given (spacial) direction relative to the reference monitor.

CYCLE_DIR

Selects the monitor in the given (cyclic) direction relative to the reference monitor.

any

Selects the first monitor that matches the given selectors.

last

Selects the previously focused monitor relative to the reference monitor.

newest

Selects the newest monitor in the history of the focused monitors.

older

Selects the monitor older than the reference monitor in the history.

newer

Selects the monitor newer than the reference monitor in the history.

focused

Selects the currently focused monitor.

pointed

Selects the monitor under the pointer.

primary

Selects the primary monitor.

^<n>

Selects the nth monitor.

<monitor_id>

Selects the monitor with the given ID.

<monitor_name>

Selects the monitor with the given name.

Modifiers*[!]focused*

Only consider the focused monitor.

[!]occupied

Only consider monitors where the focused desktop is occupied.

WINDOW STATES

tiled

Its size and position are determined by the window tree.

pseudo_tiled

A tiled window that automatically shrinks but doesn't stretch beyond its floating size.

floating

Can be moved/resized freely. Although it doesn't use any tiling space, it is still part of the window tree.

fullscreen

Fills its monitor rectangle and has no borders.

NODE FLAGS

hidden

Is hidden and doesn't occupy any tiling space.

sticky

Stays in the focused desktop of its monitor.

private

Tries to keep the same tiling position/size.

locked

Ignores the **node --close** message.

marked

Is marked (useful for deferred actions). A marked node becomes unmarked after being sent on a preselected node.

urgent

Has its urgency hint set. This flag is set externally.

STACKING LAYERS

There's three stacking layers: BELOW, NORMAL and ABOVE.

In each layer, the windows are ordered as follows: tiled & pseudo-tiled < floating < fullscreen.

RECEPTACLES

A leaf node that doesn't hold any window is called a receptacle. When a node is inserted on a receptacle in automatic mode, it will replace the receptacle. A receptacle can be inserted on a node, preselected and killed. Receptacles can therefore be used to build a tree whose leaves are receptacles. Using the appropriate rules, one can then send windows on the leaves of this tree. This feature is used in *examples/receptacles* to store and recreate layouts.

DOMAINS**Node****General Syntax**

node [*NODE_SEL*] *COMMANDS*

If *NODE_SEL* is omitted, **focused** is assumed.

Commands

-f, --focus [*NODE_SEL*]

Focus the selected or given node.

-a, --activate [*NODE_SEL*]

Activate the selected or given node.

-d, --to-desktop *DESKTOP_SEL* [**--follow**]

Send the selected node to the given desktop. If **--follow** is passed, the focused node will stay focused.

-m, --to-monitor *MONITOR_SEL* [**--follow**]

Send the selected node to the given monitor. If **--follow** is passed, the focused node will stay focused.

-n, --to-node *NODE_SEL* [**--follow**]

Send the selected node on the given node. If **--follow** is passed, the focused node will stay focused.

-s, --swap *NODE_SEL* [**--follow**]

Swap the selected node with the given node. If **--follow** is passed, the focused node will stay focused.

-p, --presel-dir [*~*]*DIR*[cancel]

Preselect the splitting area of the selected node (or cancel the preselection). If *~* is prepended to *DIR* and the current preselection direction matches *DIR*, then the argument is interpreted as

cancel. A node with a preselected area is said to be in "manual insertion mode".

- o, --presel-ratio *RATIO***
Set the splitting ratio of the preselection area.
- v, --move *dx dy***
Move the selected window by *dx* pixels horizontally and *dy* pixels vertically.
- z, --resize *top|left|bottom|right|top_left|top_right|bottom_right|bottom_left dx dy***
Resize the selected window by moving the given handle by *dx* pixels horizontally and *dy* pixels vertically.
- y, --type *CYCLE_DIR|horizontal|vertical***
Set or cycle the splitting type of the selected node.
- r, --ratio *RATIO*(+|-)(*PIXELS*|*FRACTION*)**
Set the splitting ratio of the selected node ($0 < \textit{RATIO} < 1$).
- R, --rotate *90|270|180***
Rotate the tree rooted at the selected node.
- F, --flip *horizontal|vertical***
Flip the tree rooted at selected node.
- E, --equalize**
Reset the split ratios of the tree rooted at the selected node to their default value.
- B, --balance**
Adjust the split ratios of the tree rooted at the selected node so that all windows occupy the same area.
- C, --circulate *forward|backward***
Circulate the windows of the tree rooted at the selected node.
- t, --state *~[~]STATE***
Set the state of the selected window. If *~* is present and the current state matches *STATE*, then the argument is interpreted as its last state. If the argument is just *~* with *STATE* omitted, then the state of the selected window is set to its last state.
- g, --flag *hidden|sticky|private|locked|marked[=on|off]***
Set or toggle the given flag for the selected node.
- l, --layer *below|normal|above***
Set the stacking layer of the selected window.
- i, --insert-receptacle**
Insert a receptacle node at the selected node.
- c, --close**
Close the windows rooted at the selected node.
- k, --kill**
Kill the windows rooted at the selected node.

Desktop

General Syntax

desktop [*DESKTOP_SEL*] *COMMANDS*

If *DESKTOP_SEL* is omitted, **focused** is assumed.

COMMANDS

- f, --focus [*DESKTOP_SEL*]**
Focus the selected or given desktop.

- a, --activate** [*DESKTOP_SEL*]
Activate the selected or given desktop.
- m, --to-monitor** *MONITOR_SEL* [**--follow**]
Send the selected desktop to the given monitor. If **--follow** is passed, the focused desktop will stay focused.
- s, --swap** *DESKTOP_SEL* [**--follow**]
Swap the selected desktop with the given desktop. If **--follow** is passed, the focused desktop will stay focused.
- l, --layout** *CYCLE_DIR*|monocle|tiled
Set or cycle the layout of the selected desktop.
- n, --rename** <new_name>
Rename the selected desktop.
- b, --bubble** *CYCLE_DIR*
Bubble the selected desktop in the given direction.
- r, --remove**
Remove the selected desktop.

Monitor

General Syntax

monitor [*MONITOR_SEL*] *COMMANDS*

If *MONITOR_SEL* is omitted, **focused** is assumed.

Commands

- f, --focus** [*MONITOR_SEL*]
Focus the selected or given monitor.
- s, --swap** *MONITOR_SEL*
Swap the selected monitor with the given monitor.
- a, --add-desktops** <name>...
Create desktops with the given names in the selected monitor.
- o, --reorder-desktops** <name>...
Reorder the desktops of the selected monitor to match the given order.
- d, --reset-desktops** <name>...
Rename, add or remove desktops depending on whether the number of given names is equal, superior or inferior to the number of existing desktops.
- g, --rectangle** WxH+X+Y
Set the rectangle of the selected monitor.
- n, --rename** <new_name>
Rename the selected monitor.
- r, --remove**
Remove the selected monitor.

Query

General Syntax

query *COMMANDS* [*OPTIONS*]

Commands

The optional selectors are references.

- N, --nodes** [*NODE_SEL*]
List the IDs of the matching nodes.
- D, --desktops** [*DESKTOP_SEL*]
List the IDs (or names) of the matching desktops.
- M, --monitors** [*MONITOR_SEL*]
List the IDs (or names) of the matching monitors.
- T, --tree**
Print a JSON representation of the matching item.

Options

- m, --monitor** [*MONITOR_SEL|MONITOR_MODIFIERS*], **-d, --desktop** [*DESKTOP_SEL|DESKTOP_MODIFIERS*], **-n, --node** [*NODE_SEL|NODE_MODIFIERS*]
Constrain matches to the selected monitors, desktops or nodes.
- names**
Print names instead of IDs. Can only be used with *-M* and *-D*.

Wm

General Syntax

wm *COMMANDS*

Commands

- d, --dump-state**
Dump the current world state on standard output.
- l, --load-state** <file_path>
Load a world state from the given file. The path must be absolute.
- a, --add-monitor** <name> WxH+X+Y
Add a monitor for the given name and rectangle.
- O, --reorder-monitors** <name>...
Reorder the list of monitors to match the given order.
- o, --adopt-orphans**
Manage all the unmanaged windows remaining from a previous session.
- h, --record-history** on|off
Enable or disable the recording of node focus history.
- g, --get-status**
Print the current status information.
- r, --restart**
Restart the window manager

Rule

General Syntax

rule *COMMANDS*

Commands

- a, --add** (<class_name>|*)[:(<instance_name>|*)[:(<name>|*)]] [**-o|--one-shot**]
[monitor=MONITOR_SEL|desktop=DESKTOP_SEL|node=NODE_SEL] [state=STATE]
[layer=LAYER] [honor_size_hints=(true|false|tiled|floating)] [split_dir=DIR] [split_ratio=RATIO]
[(hidden|sticky|private|locked|marked|center|follow|manage|focus|border)=(on|off)]

[rectangle=WxH+X+Y]

Create a new rule. Colons in the *instance_name*, *class_name*, or *name* fields can be escaped with a backslash.

-r, --remove ^<n>|head|tail|(<class_name>|*)[:(<instance_name>|*)[:(<name>|*)]]...

Remove the given rules.

-l, --list

List the rules.

Config

General Syntax

config [-m *MONITOR_SEL*|-d *DESKTOP_SEL*|-n *NODE_SEL*] <setting> [<value>]

Get or set the value of <setting>.

Subscribe

General Syntax

subscribe [*OPTIONS*] (all|report|monitor|desktop|node|...)*

Continuously print events. See the **EVENTS** section for the description of each event.

Options

-f, --fifo

Print a path to a FIFO from which events can be read and return.

-c, --count *COUNT*

Stop the corresponding **bspwm** process after having received *COUNT* events.

Quit

General Syntax

quit [<status>]

Quit with an optional exit status.

EXIT CODES

If the server can't handle a message, **bspwm** will return with a non-zero exit code.

SETTINGS

Colors are in the form *#RRGGBB*, booleans are *true*, *on*, *false* or *off*.

All the boolean settings are *false* by default unless stated otherwise.

Global Settings

normal_border_color

Color of the border of an unfocused window.

active_border_color

Color of the border of a focused window of an unfocused monitor.

focused_border_color

Color of the border of a focused window of a focused monitor.

presel_feedback_color

Color of the **node** **--presel-{dir,ratio}** message feedback area.

split_ratio

Default split ratio.

status_prefix

Prefix prepended to each of the status lines.

external_rules_command

Absolute path to the command used to retrieve rule consequences. The command will receive the following arguments: window ID, class name, instance name, and intermediate consequences. The

output of that command must have the following format: **key1=value1 key2=value2 ...** (the valid key/value pairs are given in the description of the *rule* command).

automatic_scheme

The insertion scheme used when the insertion point is in automatic mode. Accept the following values: **longest_side, alternate, spiral**.

initial_polarity

On which child should a new window be attached when adding a window on a single window tree in automatic mode. Accept the following values: **first_child, second_child**.

directional_focus_tightness

The tightness of the algorithm used to decide whether a window is on the *DIR* side of another window. Accept the following values: **high, low**.

removal_adjustment

Adjust the brother when unlinking a node from the tree in accordance with the automatic insertion scheme.

presel_feedback

Draw the preselection feedback area. Defaults to *true*.

borderless_monocle

Remove borders of tiled windows for the **monocle** desktop layout.

gapless_monocle

Remove gaps of tiled windows for the **monocle** desktop layout.

top_monocle_padding, right_monocle_padding, bottom_monocle_padding, left_monocle_padding

Padding space added at the sides of the screen for the **monocle** desktop layout.

single_monocle

Set the desktop layout to **monocle** if there's only one tiled window in the tree.

borderless_singleton

Remove borders of the only window on the only monitor regardless its layout.

pointer_motion_interval

The minimum interval, in milliseconds, between two motion notify events.

pointer_modifier

Keyboard modifier used for moving or resizing windows. Accept the following values: **shift, control, lock, mod1, mod2, mod3, mod4, mod5**.

pointer_action1, pointer_action2, pointer_action3

Action performed when pressing *pointer_modifier* + *button<n>*. Accept the following values: **move, resize_side, resize_corner, focus, none**.

click_to_focus

Button used for focusing a window (or a monitor). The possible values are: **button1, button2, button3, any, none**. Defaults to **button1**.

swallow_first_click

Don't replay the click that makes a window focused if *click_to_focus* isn't **none**.

focus_follows_pointer

Focus the window under the pointer.

pointer_follows_focus

When focusing a window, put the pointer at its center.

pointer_follows_monitor

When focusing a monitor, put the pointer at its center.

mapping_events_count

Handle the next **mapping_events_count** mapping notify events. A negative value implies that every

event needs to be handled.

ignore_ewmh_focus

Ignore EWMH focus requests coming from applications.

ignore_ewmh_fullscreen

Block the fullscreen state transitions that originate from an EWMH request. The possible values are: **none**, **all**, or a comma separated list of the following values: **enter**, **exit**.

ignore_ewmh_struts

Ignore strut hinting from clients requesting to reserve space (i.e. task bars).

center_pseudo_tiled

Center pseudo tiled windows into their tiling rectangles. Defaults to *true*.

remove_disabled_monitors

Consider disabled monitors as disconnected.

remove_unplugged_monitors

Remove unplugged monitors.

merge_overlapping_monitors

Merge overlapping monitors (the bigger remains).

Monitor and Desktop Settings

top_padding, *right_padding*, *bottom_padding*, *left_padding*

Padding space added at the sides of the monitor or desktop.

Desktop Settings

window_gap

Size of the gap that separates windows.

Node Settings

border_width

Window border width.

honor_size_hints

If *true*, apply ICCCM window size hints to all windows. If *floating*, only apply them to floating and pseudo tiled windows. If *tiled*, only apply them to tiled windows. If *false*, don't apply them. Defaults to *false*.

POINTER BINDINGS

click_to_focus

Focus the window (or the monitor) under the pointer if the value isn't **none**.

pointer_modifier + *button1*

Move the window under the pointer.

pointer_modifier + *button2*

Resize the window under the pointer by dragging the nearest side.

pointer_modifier + *button3*

Resize the window under the pointer by dragging the nearest corner.

The behavior of *pointer_modifier* + *button<n>* can be modified through the *pointer_action<n>* setting.

EVENTS

report

See the next section for the description of the format.

monitor_add <*monitor_id*> <*monitor_name*> <*monitor_geometry*>

A monitor is added.

monitor_rename <*monitor_id*> <*old_name*> <*new_name*>

A monitor is renamed.

monitor_remove <monitor_id>

A monitor is removed.

monitor_swap <src_monitor_id> <dst_monitor_id>

A monitor is swapped.

monitor_focus <monitor_id>

A monitor is focused.

monitor_geometry <monitor_id> <monitor_geometry>

The geometry of a monitor changed.

desktop_add <monitor_id> <desktop_id> <desktop_name>

A desktop is added.

desktop_rename <monitor_id> <desktop_id> <old_name> <new_name>

A desktop is renamed.

desktop_remove <monitor_id> <desktop_id>

A desktop is removed.

desktop_swap <src_monitor_id> <src_desktop_id> <dst_monitor_id> <dst_desktop_id>

A desktop is swapped.

desktop_transfer <src_monitor_id> <src_desktop_id> <dst_monitor_id>

A desktop is transferred.

desktop_focus <monitor_id> <desktop_id>

A desktop is focused.

desktop_activate <monitor_id> <desktop_id>

A desktop is activated.

desktop_layout <monitor_id> <desktop_id> tiled|monocle

The layout of a desktop changed.

node_add <monitor_id> <desktop_id> <ip_id> <node_id>

A node is added.

node_remove <monitor_id> <desktop_id> <node_id>

A node is removed.

node_swap <src_monitor_id> <src_desktop_id> <src_node_id> <dst_monitor_id> <dst_desktop_id>
<dst_node_id>

A node is swapped.

node_transfer <src_monitor_id> <src_desktop_id> <src_node_id> <dst_monitor_id> <dst_desktop_id>
<dst_node_id>

A node is transferred.

node_focus <monitor_id> <desktop_id> <node_id>

A node is focused.

node_activate <monitor_id> <desktop_id> <node_id>

A node is activated.

node_presel <monitor_id> <desktop_id> <node_id> (dir DIR|ratio RATIO|cancel)

A node is preselected.

node_stack <node_id_1> below|above <node_id_2>

A node is stacked below or above another node.

node_geometry <monitor_id> <desktop_id> <node_id> <node_geometry>

The geometry of a window changed.

node_state <monitor_id> <desktop_id> <node_id> tiled|pseudo_tiled|floating|fullscreen on|off

The state of a window changed.

node_flag <monitor_id> <desktop_id> <node_id> *hidden|sticky|private|locked|marked|urgent on|off*
One of the flags of a node changed.

node_layer <monitor_id> <desktop_id> <node_id> *below|normal|above*
The layer of a window changed.

pointer_action <monitor_id> <desktop_id> <node_id> *move|resize_corner|resize_side begin|end*
A pointer action occurred.

Please note that **bspwm** initializes monitors before it reads messages on its socket, therefore the initial monitor events can't be received.

REPORT FORMAT

Each report event message is composed of items separated by colons.

Each item has the form <type><value> where <type> is the first character of the item.

M<monitor_name>
Focused monitor.

m<monitor_name>
Unfocused monitor.

O<desktop_name>
Occupied focused desktop.

o<desktop_name>
Occupied unfocused desktop.

F<desktop_name>
Free focused desktop.

f<desktop_name>
Free unfocused desktop.

U<desktop_name>
Urgent focused desktop.

u<desktop_name>
Urgent unfocused desktop.

L(T|M)
Layout of the focused desktop of a monitor.

T(T|P|F|=|@)
State of the focused node of a focused desktop.

G(S?P?L?M?)
Active flags of the focused node of a focused desktop.

ENVIRONMENT VARIABLES

BSPWM_SOCKET

The path of the socket used for the communication between **bspwm** and **bspwm**. If it isn't defined, then the following path is used: */tmp/bspwm<host_name>_<display_number>_<screen_number>-socket*.

CONTRIBUTORS

- Steven Allen <steven at stebalien.com>
- Thomas Adam <thomas at xteddy.org>
- Ivan Kanakarakis <ivan.kanak at gmail.com>

AUTHOR

Bastien Dejean <nihilhill at gmail.com>