broot Tree navigator

START HERE

br
Start broot (shell function)

Esc
Clear filter / go back

<letters>
Fuzzy-filter the tree

?
Open the help screen

Enter
Open file / focus directory

:cp *<path>*
Copy the selection

Alt+Enter
Leave and cd to directory

Ctrl+q
Quit

CONCEPTS

Run broot through the br function, not the binary: only then can Alt+Enter change your shell's directory.

Pattern letters, /regex, c/text
↓ narrows
Tree selected line
↓ space or :
Verb :cp, :rm, :e
↓ acts on
Selection file or staged set

Every keystroke in the input is one of these

Type a pattern first, then a verb: the input takes both, e.g. 'readme :e'. Esc clears the verb, a second Esc clears the pattern.

LAUNCH

br
Start broot in the current directory

br *<path>*
Start in another directory, or preview a file

br -s
Show sizes

br -h
Show hidden files

br -i
Show git-ignored files

br -d
Show last-modified dates

br -p
Show permissions, owner and group

br -sdp
Use as an ls replacement
Flags combine; an uppercase flag (-S, -H, -I) turns the setting off.

br -w
Whale mode: find what eats disk space

br -f
Show folders only

br -g
Show git status of files

br -gg
Only files with a git status

br -c "*<cmds>***"**
Run commands as if typed, separated by ;

broot --install
(Re)install the br shell function

NAVIGATE

Arrows
Move the selection

NAVIGATE (cont.)

Enter
Open file / make directory the root

Alt+Enter
Leave broot, cd the shell to the directory
Needs the br function. On a file it leaves broot and opens the file.

Esc
Clear input / return to previous state / quit

?
Show help and config file location

Ctrl+q
Quit

:parent
Focus the parent directory

:focus *<path>*
Make a directory the root

:focus ~
Jump to your home directory

Ctrl+s
Search the whole tree / restore last pattern

Alt+h
Toggle hidden files

Alt+i
Toggle git-ignored files

SEARCH

<letters>
Fuzzy match on sub-path from the root

n/*<text>*
Fuzzy match on file name only

e/*<text>*
Exact match on file name

/*<regex>*
Regex on file name

/*<regex>***/i**
Case-insensitive regex

c/*<text>*
Files containing text

rc/*<regex>*
Files whose content matches regex

!*<pattern>*
Negate a pattern

*<a>***&***<b>*
Both patterns must match

*<a>***|***<b>*
Either pattern matches

VIEW

:sort
Cycle sort: date, count, size, type
Press Tab after :sort to rotate through the options.

:size
Toggle sizes

:dates
Toggle dates

:perm
Toggle permissions

:gf
Toggle git status per file

:gs
Show only files git sees as changed

:gd
Diff the selected file

:fs
Show filesystems and disk usage

FILES

:e
Edit file in your editor
Opens at the matching line after a content search.

:cp *<path>*
Copy the selection

:mv *<path>*
Move the selection

:rm
Delete the selection

:trash
Send the selection to the trash

:rename
Rename (F2), pre-filled with current name

:md *<name>*
Create a directory (slashes allowed)

:restore_trashed_file
Put a trashed file back

PANELS AND STAGING

Ctrl+Right
Open or focus the panel on the right
Alt+Right on macOS. Right on a file focuses its preview.

Ctrl+Left
Open or focus the left panel

Ctrl+w
Close the current panel

:focus! *<path>*
Open a directory in a new panel
A trailing or leading ! sends any verb's result to a new panel.

PANELS AND STAGING (cont.)

:cpp
Copy selection to the other panel

:mvp
Move selection to the other panel

Ctrl+g
Stage or unstage the selection

Ctrl+a
Stage all files of the left panel

:cls
Clear the staging area

:preview_diff
Preview the git diff of a file

OUTPUT AND CONFIG

:pt
Print the tree to stdout and quit

:print_path
Print the selected path and quit

br --cmd ":pt" > *<file>*
Export the tree to a file

conf.hjson
Config file in ~/.config/broot
TOML works too (conf.toml). Press ? in broot to see the exact path.

default_flags: "-gh"
Always start with these launch flags

RECIPES

Open a file in nvim from broot

```
# conf.hjson
verbs: [
  {
    invocation: edit
    key: Enter
    shortcut: e
    external: "nvim +{line}
{file}"
    apply_to: text_file
    leave_broot: false
  }
]
```

Deep cd with a pattern

```
# bash/zsh, in your rc file
function dcd { br --only-
folders --cmd "$1;:cd"; }
dcd <pattern>
```

Replace tree

```
function tree { br -c :pt
"$@"; }
tree -s ~/projects
```

Review a repo's changes

```
br -ghc :gs
# select a file, then
# :gd to see its diff
# Esc to return to the full
tree
```