

# FD

## NAME

fd - find entries in the filesystem

## SYNOPSIS

**fd** [-HIEsiaLp0hV] [-d *depth*] [-t *filetype*] [-e *ext*] [-E *exclude*] [-c *when*] [-j *num*] [-x *cmd*] [*pattern*] [*path...*]

## DESCRIPTION

**fd** is a simple, fast and user-friendly alternative to **find**(1).

By default **fd** uses regular expressions for the pattern. However, this can be changed to use simple glob patterns with the **'--glob'** option.

By default **fd** will exclude hidden files and directories, as well as any files that match gitignore rules or ignore rules in .ignore or .fdignore files.

## OPTIONS

### -H, --hidden

Include hidden files and directories in the search results (default: hidden files and directories are skipped). The flag can be overridden with **'--no-hidden'**.

Ignored files are still excluded unless **--no-ignore** or **--no-ignore-vcs** is also used.

### -I, --no-ignore

Show search results from files and directories that would otherwise be ignored by

- *.gitignore*
- *.git/info/exclude*
- The global gitignore configuration (by default *\$HOME/.config/git/ignore*)
- *.ignore*
- *.fdignore*
- The global fd ignore file (usually *\$HOME/.config/fd/ignore*)

The flag can be overridden with **'--ignore'**.

### -u, --unrestricted

Perform an unrestricted search, including ignored and hidden files. This is an alias for **'--hidden --no-ignore'**.

### --no-ignore-vcs

Show search results from files and directories that would otherwise be ignored by gitignore files including *.gitignore*, *.git/info/exclude*, and the global gitignore configuration (*core.excludesFile* git setting, which defaults to *\$HOME/.config/git/ignore*). The flag can be overridden with **'--ignore-vcs'**.

### --no-require-git

Do not require a git repository to respect gitignores. By default, fd will only respect global gitignore rules, .gitignore rules and local exclude rules if fd detects that you are searching inside a git repository. This flag allows you to relax this restriction such that fd will respect all git related ignore rules regardless of whether you're searching in a git repository or not. The flag can be overridden with **'--require-git'**.

### --no-ignore-parent

Show search results from files and directories that would otherwise be ignored by gitignore files in parent directories.

### -s, --case-sensitive

Perform a case-sensitive search. By default, fd uses case-insensitive searches, unless the pattern contains an uppercase character (smart case).

### -i, --ignore-case

Perform a case-insensitive search. By default, fd uses case-insensitive searches, unless the pattern contains an uppercase character (smart case).

### -g, --glob

Perform a glob-based search instead of a regular expression search. If combined with the **'--full-path'** option, **'\*\*'** can be used to match multiple path components.

### --regex

Perform a regular-expression based search (default). This can be used to override **--glob**.

### -F, --fixed-strings

Treat the pattern as a literal string instead of a regular expression. Note that this also performs substring comparison. If you want to match on an exact filename, consider using **'--glob'** or **'--fixed-strings'** is used.

### --and pattern

Add additional required search patterns, all of which must be matched. Multiple additional patterns can be specified. The patterns are regular expressions, unless **'--glob'** or **'--fixed-strings'** is used.

### -a, --absolute-path

Shows the full path starting from the root as opposed to relative paths. The flag can be overridden with **'--relative-path'**.

### -l, --list-details

Use a detailed listing format like **'ls -l'**. This is basically an alias for **'--exec-batch ls -l'** with some additional **'ls'** options. This can be used to see more metadata, to show symlink targets and to achieve a deterministic sort order.

### -L, --follow

By default, fd does not descend into symlinked directories. Using this flag, symbolic links are also traversed. The flag can be overridden with **'--no-follow'**.

### -p, --full-path

By default, the search pattern is only matched against the filename (or directory name). Using this flag, the *pattern* is matched against the full path.

### -0, --print0

Separate search results by the null character (instead of newlines). Useful for piping results to *xargs*.

### --max-results count

Limit the number of search results to **'count'** and quit immediately.

**-1**

Limit the search to a single result and quit immediately. This is an alias for **'--max-results=1'**.

### -q, --quiet

When the flag is present, the program does not print anything and will instead exit with a code of 0 if there is at least one search result. Otherwise, the exit code will be 1. This is mainly for usage in scripts and can be faster than checking for output because the search can be stopped early after the first match. **--has-results** can be used as an alias.

### --show-errors

Enable the display of filesystem errors for situations such as insufficient permissions or dead symlinks.

### --strip-cwd-prefix [when]

By default, relative paths are prefixed with **'./'** when **-x/--exec**, **-X/--exec-batch**, or **-0/--print0** are given, to reduce the risk of a path starting with **'.'** being treated as a command line option. Use this flag to change this behavior. If this flag is used without a value, it is equivalent to passing "always". Possible values are:

never	Never strip the <b>./</b> at the beginning of paths
always	Always strip the <b>./</b> at the beginning of paths
auto	Only strip if used with <b>--exec</b> , <b>--exec-batch</b> , or <b>--print0</b> . That is, it resets to the default behavior.

### --one-file-system, --mount, --xdev

By default, fd will traverse the file system tree as far as other options dictate. With this flag, fd ensures that it does not descend into a different file system than the one it started in. Comparable to the **-mount** or **-xdev** filters of **find**(1).

### -h, --help

Print help information.

### -V, --version

Print version information.

### -d, --max-depth d

Limit directory traversal to at most **d** levels of depth. By default, there is no limit on the search depth.

### --min-depth d

Only show search results starting at the given depth. See also: **'--max-depth'** and **'--exact-depth'**.

### --exact-depth d

Only show search results at the exact given depth. This is an alias for **'--min-depth <depth> --max-depth <depth>'**.

### --prune

Do not traverse into matching directories.

### -t, --type filetype

- *f*, *file* regular files
- *d*, *dir*, *directory* directories
- *l*, *symlink* symbolic links
- *b*, *block-device* block devices
- *c*, *char-device* character devices
- *s*, *socket* sockets
- *p*, *pipe* named pipes (FIFOs)
- *x*, *executable* executable (files)
- *e*, *empty* empty files or directories

This option can be specified more than once to include multiple file types. Searching for **'--type file --type symlink'** will show both regular files as well as symlinks. Note that the **'executable'** and **'empty'** filters work differently: **'--type executable'** implies **'--type file'** by default. And **'--type empty'** searches for empty files and directories, unless either **'--type file'** or **'--type directory'** is specified in addition.

Examples: - Only search for files: **fd --type file ... fd -tf ...** - Find both files and symlinks **fd --type file --type symlink ... fd -tf -tl ...** - Find executable files: **fd --type executable fd -tx** - Find empty files: **fd --type empty --type file fd -te -tf** - Find empty directories: **fd --type empty --type directory fd -te -td**

### -e, --extension ext

Filter search results by file extension *ext*. This option can be used repeatedly to allow for multiple possible file extensions.

If you want to search for files without extension, you can use the regex **'^[^.]+'\$** as a normal search pattern.

### -E, --exclude pattern

Exclude files/directories that match the given glob pattern. This overrides any other ignore logic. Multiple exclude patterns can be specified. Examples: **--exclude \*.pyc** **--exclude node\_modules**

### --ignore-file path

Add a custom ignore-file in **'gitignore'** format. These files have a low precedence.

### -c, --color when

Declare *when* to colorize search results:

auto	Colorize output when standard output is connected to terminal (default).
never	Do not colorize output.
always	Always colorize output.

### --hyperlink

Specify whether the output should use terminal escape codes to indicate a hyperlink to a file url pointing to the path. The value can be **auto**, **always**, or **never**.

Currently, the default is "never", and if the option is used without an argument "auto" is used. In the future this may be changed to "auto" and "always".

auto	Only output hyperlinks if color is also enabled, as a proxy for whether terminal escape codes are acceptable.
never	Never output hyperlink escapes.
always	Always output hyperlink escapes, regardless of color settings.

### -j, --threads num

Set number of threads to use for searching & executing (default: number of available CPU cores).

**-S, --size size**

Limit results based on the size of files using the format `<+-><NUM><UNIT>`

'+'	file size must be greater than or equal to this
'.'	file size must be less than or equal to this

If neither '+' nor '.' is specified, file size must be exactly equal to this.

NUM	The numeric size (e.g. 500)
'UNIT'	The units for NUM. They are not case-sensitive. Allowed unit values:

'b'	bytes
'k'	kilobytes (base ten, 10 <sup>3</sup> = 1000 bytes)
'm'	megabytes
'g'	gigabytes
't'	terabytes
'ki'	kibibytes (base two, 2 <sup>10</sup> = 1024 bytes)
'mi'	mebibytes
'gi'	gibibytes
'ti'	tebibytes

**--changed-within date|duration**

Filter results based on the file modification time. Files with modification times greater than the argument will be returned. The argument can be provided as a duration (**10h, 1d, 35min**) or as a specific point in time as full RFC3339 format with time zone, as a date or datetime in the local time zone (**YYYY-MM-DD** or **YYYY-MM-DD HH:MM:SS**), or as the prefix '@' followed by the number of seconds since the Unix epoch (@[0-9]+). **--change-newer-than**, **--newer** or **--changed-after** can be used as aliases.

Examples: **--changed-within 2weeks** **--change-newer-than "2018-10-27 10:00:00"** **--newer 2018-10-27** **--changed-after @1704067200**

**--changed-before date|duration**

Filter results based on the file modification time. Files with modification times less than the argument will be returned. The argument can be provided as a duration (**10h, 1d, 35min**) or as a specific point in time as full RFC3339 format with time zone, as a date or datetime in the local time zone (**YYYY-MM-DD** or **YYYY-MM-DD HH:MM:SS**), or as the prefix '@' followed by the number of seconds since the Unix epoch (@[0-9]+). **--change-older-than** or **--older** can be used as aliases.

Examples: **--changed-before "2018-10-27 10:00:00"** **--change-older-than 2weeks** **--older @1704067200**

**-o, --owner [user][:group]**

Filter files by their user and/or group. Format: [(user[uid])][: (group[gid])]. Either side is optional. Precede either side with a '!' to exclude files instead.

Examples: **--owner john** **--owner :students** **--owner "john:students"**

**--base-directory path**

Change the current working directory of fd to the provided path. This means that search results will be shown with respect to the given base path. Note that relative paths which are passed to fd via the positional **path** argument or the **--search-path** option will also be resolved relative to this directory.

**--path-separator separator**

Set the path separator to use when printing file paths. The default is the OS-specific separator ('/' on Unix, '\' on Windows).

**--search-path search-path**

Provide paths to search as an alternative to the positional **path** argument. Changes the usage to 'fd [FLAGS/OPTIONS] --search-path PATH --search-path PATH2 [PATTERN]'

**--format fmt**

Specify a template string that is used for printing a line for each file found. The following placeholders are substituted into the string for each file before printing:

{}	path (of the current search result)
{/}	basename
{//}	parent directory
{.}	path without file extension
{/}	basename without file extension
{{	literal '{' (an escape sequence)
}}	literal '}' (an escape sequence)

Notice that you can use "{{" and "}}" to escape "{" and "}" respectively, which is especially useful if you need to include the literal text of one of the above placeholders.

**-x, --exec command**

Execute **command** for each search result in parallel (use **--threads=1** for sequential command execution).

Note that all subsequent positional arguments are considered to be arguments to the **command** - not to fd. It is therefore recommended to place the **-x/--exec** option last. Alternatively, you can supply a ';' argument to end the argument list and continue with more fd options. Most shells require ';' to be escaped: '\;'. This option can be specified multiple times, in which case all commands are run for each file found, in the order they are provided. In that case, you must supply a ';' argument for all but the last commands.

If parallelism is enabled, the order commands will be executed in is non-deterministic. And even with **--threads=1**, the order is determined by the operating system and may not be what you expect. Thus, it is recommended that you don't rely on any ordering of the results.

Before executing the command, any placeholder patterns in the command are replaced with the corresponding values for the current file. The same placeholders are used as in the **--format** option.

If no placeholder is present, an implicit "{}" at the end is assumed.

Examples:

- find all \*.zip files and unzip them:  
fd -e zip -x unzip
- find \*.h and \*.cpp files and run "clang-format -i .." for each of them:  
fd -e h -e cpp -x clang-format -i
- Convert all \*.jpg files to \*.png files:

fd -e jpg -x convert {} {}.png

**-X, --exec-batch command**

Execute **command** once, with all search results as arguments.

The order of the arguments is non-deterministic and should not be relied upon.

This uses the same placeholders as **--format** and **--exec**, but instead of expanding once per command invocation each argument containing a placeholder is expanding for every file in a batch and passed as separate arguments.

If no placeholder is present, an implicit "{}" at the end is assumed.

Like **--exec**, this can be used multiple times, in which case each command will be run in the order given.

Examples:

- Find all test \*.py files and open them in your favorite editor:  
fd -g 'test \*.py' -X vim
- Note that this executes a single "vim" process with all search results as arguments.
- Find all \*.rs files and count the lines with "wc -l ...":  
fd -e rs -X wc -l

**--batch-size size**

Maximum number of arguments to pass to the command given with -X. If the number of results is greater than the given size, the command given with -X is run again with remaining arguments. A batch size of zero means there is no limit (default), but note that batching might still happen due to OS restrictions on the maximum length of command lines.

**PATTERN SYNTAX**

The regular expression syntax used by fd is documented here: <https://docs.rs/regex/1.0.0/regex/#syntax>

The glob syntax is documented here: <https://docs.rs/globset/#syntax>

**ENVIRONMENT**

**LS\_COLORS**

Determines how to colorize search results, see **dircolors(1)**.

**NO\_COLOR**

Disables colorized output.

**XDG\_CONFIG\_HOME, HOME**

Used to locate the global ignore file. If **XDG\_CONFIG\_HOME** is set, use **\$XDG\_CONFIG\_HOME/fd/ignore**. Otherwise, use **\$HOME/.config/fd/ignore**.

**FILES**

**.fdignore**

This file works similarly to a .gitignore file anywhere in the searched tree and specifies patterns that should be excluded from the search. However, this file is specific to fd, and will be used even if the **--no-ignore-vcs** option is used.

**\$XDG\_CONFIG\_HOME/fd/ignore**

Global ignore file. Unless ignore mode is turned off (such as with **--no-ignore**) ignore entries in this file will be ignored, as if it was an .fdignore file in the current directory.

**EXAMPLES**

Find all Python files (all files with the extension **.py**) in the current directory:

\$ fd -e py

Open all search results with vim:

\$ fd pattern -X vim

**Tips and Tricks**

- If you add ".git/" to your global ignore file (\$XDG\_CONFIG\_HOME/fd/ignore), then ".git" folders will be ignored by default, even when the **--hidden** option is used.
- You can use a shell alias or a wrapper script in order to pass desired flags to fd by default. For example if you do not like the default behavior of respecting gitignore, you can use 'alias fd="/usr/bin/fd --no-ignore-vcs"' in your .bashrc to create an alias for fd that doesn't ignore git files by default.

**BUGS**

Bugs can be reported on GitHub: <https://github.com/sharkdp/fd/issues>

**SEE ALSO**

**find(1)**