

Why use fselect?

While it doesn't tend to fully replace traditional `find` and `ls`, **fselect** has these nice features:

- SQL-like (not real SQL, but highly relaxed!) grammar easily understandable by humans
- complex queries
- aggregate, statistics, date, and other functions
- search within archives
- `.gitignore`, `.hgignore`, and `.dockerignore` support (experimental)
- search by width and height of images, EXIF metadata
- search by MP3 info
- search by extended file attributes
- search by file hashes
- search by MIME type
- shortcuts to common file types
- interactive mode
- various output formatting (CSV, JSON, and others)

More is under way!

Installation

Latest release from source

- Install [Rust with Cargo](#) and its dependencies to build a binary
- Run `cargo install fselect`

Arch Linux

[AUR package](#), thanks to [@asm0dey](#)

NixOS

[fselect in nixpkgs](#), thanks to [@filalex77](#)

Other Linux

[Static build with musl](#).

Windows 64bit

A statically precompiled [binary](#) is available at Github downloads.

Windows via winget

- Install [winget](#)
- Run `winget install -e --id fselect.fselect`

Windows via Chocolatey

- Install [Chocolatey](#)
- Run `choco install fselect`

Windows via Scoop

- Install [Scoop](#)
- Run `scoop install fselect`

Mac via Homebrew

- Install [brew](#)
- Run `brew install fselect`

Mac via MacPorts

- Install [MacPorts](#)
- Run:

```
sudo port selfupdate
sudo port install fselect
```



Usage

```
fselect [ARGS] COLUMN[, COLUMN...] [from ROOT[, ROOT...]] [where EXPR] [group by COLUMNS]
[order by COLUMNS] [limit N] [into FORMAT]
```



Interactive mode

```
fselect -i
```



Documentation

[More detailed description. Look at examples first.](#)

Examples

Find temporary or config files (full path and size):

```
fselect size, path from /home/user where name = '*.cfg' or name = '*.tmp'
```



Windows users may omit the quotes:

```
fselect size, path from C:\Users\user where name = *.cfg or name = *.tmp
```



Or put all the arguments into the quotes like this:

```
fselect "name from /home/user/tmp where size > 0"
```



Search within a directory name with spaces (backticks are also supported):

```
fselect "name from '/home/user/dir with spaces' where size > 0"  
fselect "name from `/home/user/dir with spaces` where size > 0"
```



Or simply escape the single quotes:

```
fselect name from \"'/home/user/dir with spaces'\" where size gt 0
```



Specify file size, get absolute path, and add it to the results:

```
cd /home/user  
fselect size, abspath from ./tmp where size gt 2g  
fselect fsize, abspath from ./tmp where size = 5m  
fselect hsize, abspath from ./tmp where size lt 8k  
fselect name, size from ./tmp where size between 5mb and 6mb
```



More complex query:

```
fselect "name from /tmp where (name = *.tmp and size = 0) or (name = *.cfg and size > 1000000)"
```



Aggregate functions (you can use curly braces if you want, and even combine them with the regular parentheses):

```
fselect "MIN(size), MAX{size}, AVG(size), SUM{size}, COUNT(*) from /home/user/Downloads"
```



Formatting functions:

```
fselect "LOWER(name), UPPER(name), LENGTH(name), YEAR(modified) from /home/user/Downloads"
```



Get the year of an oldest file:

```
fselect "MIN(YEAR(modified)) from /home/user"
```



Use single quotes if you need to address files with spaces:

```
fselect "path from '/home/user/Misc stuff' where name != 'Some file'"
```



Regular expressions of [Rust flavor](#) are supported:

```
fselect name from /home/user where path =~ '.*Rust.*'
```



Negate regular expressions:

```
fselect "name from . where path !=~ '^\.\/config'"
```



Simple globs expand automatically and work with `=` and `!=` operators:

```
fselect name from /home/user where path = '*Rust*'
```



Classic LIKE:

```
fselect "path from /home/user where name like '%report-2018-__-__???'"
```



Exact match operators to search with regexps disabled:

```
fselect "path from /home/user where name === 'some_*_weird_*_name'"
```



Find files by date:

```
fselect path from /home/user where created = 2017-05-01
fselect path from /home/user where modified = today
fselect path from /home/user where accessed = yesterday
fselect "path from /home/user where modified = 'apr 1'"
fselect "path from /home/user where modified = 'last fri'"
```



Be more specific to match all files created at interval between 3PM and 4PM:

```
fselect path from /home/user where created = '2017-05-01 15'
```



And even more specific:

```
fselect path from /home/user where created = '2017-05-01 15:10'
fselect path from /home/user where created = '2017-05-01 15:10:30'
```



Date and time intervals possible (find everything updated since May 1st):

```
fselect path from /home/user where modified gte 2017-05-01
```



Default is current directory:

```
fselect path, size where name = '*.jpg'
```



Search within multiple locations:

```
fselect path from /home/user/oldstuff, /home/user/newstuff where name = '*.jpg'
```



With minimum and/or maximum depth specified (`depth` is a synonym for `maxdepth`):

```
fselect path from /home/user/oldstuff depth 5 where name = '*.jpg'
fselect path from /home/user/oldstuff mindepth 2 maxdepth 5, /home/user/newstuff depth 10 where
name = '*.jpg'
```



Optionally follow symlinks:

```
fselect path, size from /home/user symlinks where name = '*.jpg'
```



Search within archives (currently only zip-archives are supported):

```
fselect path, size from /home/user archives where name = '*.jpg'
```



Or in combination:

```
fselect size, path from /home/user depth 5 archives symlinks where name = '*.jpg' limit 100
```



Enable `.gitignore` or `.hgignore` support:

```
fselect size, path from /home/user/projects gitignore where name = '*.cpp'
fselect size, path from /home/user/projects hgignore where name = '*.py'
```



Search by image dimensions:

```
fselect CONCAT(width, 'x', height), path from /home/user/photos where width gte 2000 or height
gte 2000
```



Find square images:

```
fselect path from /home/user/Photos where width = height
```



Find images with a known name part but unknown extension:

```
fselect path from /home/user/projects where name = "*RDS*" and width gte 1
```



Find old-school rap MP3 files:

```
fselect duration, path from /home/user/music where genre = Rap and bitrate = 320 and mp3_year
lt 2000
```



Shortcuts to common file extensions:

```
fselect path from /home/user where is_archive = true
fselect path, mime from /home/user where is_audio = 1
```



```
fselect path, mime from /home/user where is_book != false
```

Even simpler way of using boolean columns:

```
fselect path from /home/user where is_doc  
fselect path from /home/user where is_image  
fselect path from /home/user where is_video
```

Find files with dangerous permissions:

```
fselect mode, path from /home/user where other_write or other_exec  
fselect mode, path from /home/user where other_all
```

Simple glob-like expressions or even regular expressions on file mode are possible:

```
fselect mode, path from /home/user where mode = '*rwx'  
fselect mode, path from /home/user where mode =~ '.*rwx$'
```

Find files by owner's uid or gid:

```
fselect uid, gid, path from /home/user where uid != 1000 or gid != 1000
```

Or by owner's or group's name:

```
fselect user, group, path from /home/user where user = mike or group = mike
```

Find special files:

```
fselect name from /usr/bin where suid  
fselect path from /tmp where is_pipe  
fselect path from /tmp where is_socket
```

Find files with xattrs, check if particular xattr exists, or get its value:

```
fselect "path, has_xattrs, has_xattr(user.test), xattr(user.test) from /home/user"
```

Include arbitrary text as columns:

```
fselect "name, ' has size of ', size, ' bytes'"
```

Group results:

```
fselect "ext, count(*) from /tmp group by ext"
```

Order results:

```
fselect path from /tmp order by size desc, name
fselect modified, fsize, path from ~ order by 1 desc, 3
```



Finally limit the results:

```
fselect name from /home/user/samples limit 5
```



Format output:

```
fselect size, path from /home/user limit 5 into json
fselect size, path from /home/user limit 5 into csv
fselect size, path from /home/user limit 5 into html
```



License

MIT/Apache-2.0

Releases 33

 **0.8.6** Latest
on May 3

[+ 32 releases](#)

Packages

No packages published

Contributors 21




[+ 7 contributors](#)

Deployments 157

 **github-pages** last week

[+ 156 deployments](#)

Languages

 Rust 100.0%