
Count Files Documentation

Release 1.6.0

Victor Domingos & Nataliia Bondarenko

Jul 04, 2020

Contents

1	Installation and dependencies	3
1.1	On regular desktop operating systems	3
1.2	On iPhone or iPad (in Pythonista 3 for iOS)	3
2	How to use	5
2.1	CLI arguments	5
2.2	Getting help	6
2.3	The <code>path</code> argument	6
2.4	Non-recursive search or counting	6
2.5	Hidden files and directories	7
2.6	Case sensitivity	7
2.7	Customizing operation feedback	7
2.8	File counting by extension	7
2.9	The <code>--sort-alpha</code> argument	8
2.10	File searching by extension	8
2.11	File searching by pattern	8
2.12	Total counting of files	8
2.13	Preview text files	9
2.14	File sizes	9
3	Examples of practical usage	11
3.1	File counting by extension: sorted table	11
3.2	File searching by extension: list with file paths	13
3.3	File searching by pattern: list with file paths	15
3.4	Total number of files	17
4	Search	19

A command-line interface (CLI) utility written in Python to help you count files, grouped by extension, in a directory. By default, it will count files recursively in current working directory and all of its subdirectories, and will display a table showing the frequency for each file extension (e.g.: .txt, .py, .html, .css) and the total number of files found.

Main functions:

- counting files by extension and sorting results by frequency or alphabetically;
- file searching by extension, additionally: file sizes, preview for text files;
- file searching by pattern, additionally: file sizes, preview for text files;
- total counting of all files or files with a specific extension, optional: list of folders in which found files are located, number and total combined size of files found.

Features:

- counting or searching files with a specific extension, files without an extension, all files regardless of the extension;
- recursive and non-recursive counting or searching;
- an option to include or exclude hidden files and folders while searching or counting;
- an option to process the file extensions with case-sensitive or case-insensitive mode;
- an option to turn off the program's operating indicator (feedback).

Supported operating systems:

Linux, macOS, Windows.

It may also be used on iOS (iPhone/iPad) using the [StaSh](#) command-line in the Pythonista 3 app.

Supported Python versions:

Python 3.6 and later

Tested:

- Linux Mint 19 "Tara", Python 3.6.5
- Windows 8.1 Professional, Python 3.6.0
- macOS 10.13.6 High Sierra, Python 3.7.0
- macOS 10.13.6 High Sierra, Python 3.6.6
- macOS 12.12.6 Sierra, Python 3.7.0
- macOS 12.12.6 Sierra, Python 3.6.6
- macOS 12.11.6 El Capitan, Python 3.7.0
- macOS 12.11.6 El Capitan, Python 3.6.6
- iOS 12.0 (Pythonista 3.2), Python 3.6.1
- iOS 9.3.5 (Pythonista 3.2), Python 3.6.1

License: MIT

Source on GitHub: [Project repository](#)

Contents:

Installation and dependencies

1.1 On regular desktop operating systems

Count Files is a platform-independent application that runs in Python and can be easily installed using `pip`:

```
pip3 install count-files
```

If you are interested in the current development version, you can simply clone this git repository and install it using `pip3 install -e`. Please notice, however, that only released versions are expected to be stable and usable. The development code is often unstable or buggy, for the simple reason that it is a work in progress.

1.2 On iPhone or iPad (in Pythonista 3 for iOS)

First you will need a Python environment and a command-line shell compatible with Python 3. Presently, it means you need to have an app called `Pythonista 3` (which is, among other things, a very nice environment for developing and/or running pure Python applications on iOS).

Then you need to install `StaSh`, which is a Python-based shell application for Pythonista. It will enable you to use useful commands like `wget`, `git clone`, `pip install` and many others. It really deserves an home screen shortcut on your iPhone or iPad.

After following the instructions for `StaSh` installation, you may need to update it to a more recent version. Try this command:

```
selfupdate.py -f dev
```

Then force-quit and restart Pythonista and launch `StaSh` again. It should now be running in Python 3. You may now try to install this application, directly from this git repository:

```
pip install count-files
```

If all goes well, it should place a new `count_files` package inside the `~/Documents/site-packages-3/` folder and create an entry point script named `count-files.py` in `stash_extensions/bin`. Then force-quit and launch StaSh again. You should now be able to run this application directly from the shell to count any files that you may have inside your Pythonista environment.

If you are interested in the current development version, you can clone this git repository into your Pythonista environment using StaSh. You can also install it directly using:

```
pip3 install victordomingos/Count-files
```

Please notice, however, that only released versions are expected to be stable and usable. The development code is often unstable or buggy, for the simple reason that it is a work in progress.

2.1 CLI arguments

Arguments can be specified in both short and long form. For example: `-a` or `--all`.

```
usage: count-files [-h] [-hc]
                  [-v] [-st]
                  [-a] [-c]
                  [-nr] [-nf]
                  [-alpha]
                  [-t EXTENSION]
                  [-sf] [-ts]
                  [-fe FILE_EXTENSION]
                  [-fm PATTERN] [-fs]
                  [-p] [-ps PREVIEW_SIZE]
                  [path]

usage: count-files [--help] [--help-cmd]
                  [--version] [--supported-types]
                  [--all] [--case-sensitive]
                  [--no-recursion] [--no-feedback]
                  [--sort-alpha]
                  [--total EXTENSION]
                  [--show-folders] [--total-size]
                  [--file-extension FILE_EXTENSION]
                  [--filename-match PATTERN] [--file-sizes]
                  [--preview] [--preview-size PREVIEW_SIZE]
                  [path]
```

2.1.1 Common arguments

`path`, `--all`, `--case-sensitive`, `--no-recursion`, `--no-feedback`

2.1.2 Special arguments

- **File counting by extension (sorted table):** `--sort-alpha`
- **File searching by extension (list with file paths):** `--file-extension`, `--file-sizes`, `--preview`, `--preview-size`
- **File searching by pattern (list with file paths):** `--filename-match`, `--file-sizes`, `--preview`, `--preview-size`
- **Total number of files (number):** `--total`, `--show-folders`, `--total-size`

2.2 Getting help

To check the list of available options and their usage, you just need to use one of the following commands:

```
count-files -h  
count-files --help
```

Search in help by topic - argument or group name(count, search, total):

```
count-files -hc  
count-files --help-cmd
```

Check the version number of the program:

```
count-files -v  
count-files --version
```

Get the list of currently supported file types for preview:

```
count-files -st  
count-files --supported-types
```

2.3 The path argument

Optionally, you can pass it a path to the directory to scan. If you prefer, you can leave that argument empty, and it will scan the current working directory.

To process files in the user's home directory, you can use `~` (tilde).

If there are spaces in the folder names, then path should be specified in quotation marks. For example, in Windows:
`count-files "~\Desktop\New folder"`

2.4 Non-recursive search or counting

The optional `-nr` or `--no-recursion` switch argument tells the application not to scan recursively through the subdirectories.

2.5 Hidden files and directories

For supported operating systems (Linux, macOS, iOS, Windows), any hidden files or folders are ignored by default. But you can add the `-a` or `--all` optional switch argument to make it count or search for all files.

For other operating systems in which Python can be run, this option to include/exclude hidden files is not available. And as a result, all existing files will be included.

In Windows, files and directories considered hidden by this application are those for which the `FILE_ATTRIBUTE_HIDDEN` attribute is set to true.

In Linux, macOS, iOS and other Unix-like operating systems, a file or directory is considered to be hidden if its name starts with a `.` (dot).

2.6 Case sensitivity

The names of extensions are case insensitive by default. The results for `ini` and `INI` will be the same. To distinguish between similar extensions in different cases, use the `-c` or `--case-sensitive` switch argument.

- File counting by extension (sorted table):

In this case, the file extensions in the table will be displayed as is (in lowercase and uppercase).

- File searching by extension (using `-fe` or `--file-extension`):

The result of the search will be a list with paths to files with an extension in the corresponding register.

- File searching by pattern (using `-fm` or `--filename-match`):

The search result will be a list of file paths with file names that exactly match the case-sensitive pattern.

- Total number of files (using `-t` or `--total`):

For total counting of files with a specific extension, this option is also available. The result of the counting will be a total number of files with an extension in the corresponding register.

2.7 Customizing operation feedback

By default, the program displays an operating indicator that consists in successively printing processed file names in a single line. File names are not displayed, however, when searching for a particular extension, if there are no such files in the folder or if the files are hidden, and the argument `--all` was not specified.

This kind of feedback is available by default when counting files by extension and when counting the total number of files (using `-t` or `--total`). The optional `-nf` or `--no-feedback` switch argument disables it.

Using the `--no-feedback` argument allows you to speed up a little the processing of a large amount of files/folders.

When searching for files by extension or by pattern (using `--file-extension` or `--filename-match`) the feedback mechanism is the list of file paths itself.

2.8 File counting by extension

To count all files by extension, you can simply use the command `count-files` and, if necessary, specify one or more of the common arguments: `path`, `--all`, `case-sensitive`, `--no-recursion`, `--no-feedback`.

You can sort the extensions in the table alphabetically using the `--sort-alpha` argument.

Example: `count-files`

See also:

File counting by extension: sorted table

2.9 The `--sort-alpha` argument

By default, result of file counting by extension is a table that lists all the file extensions found and displays the frequency for each file extension. To sort the extensions alphabetically, use the `-alpha` or `--sort-alpha` argument.

Example: `count-files --sort-alpha`

2.10 File searching by extension

Another main feature of this application consists in searching files by a given extension, which presents to the user a list of all found files.

Using `-fe` or `--file-extension` argument, you can find all the files that have the specified extension. You can get additional information about the size of each found file and see a short preview for text files (`--file-sizes`, `--preview`, `--preview-size` optional arguments).

If necessary, specify one or more of the common arguments: `path`, `--all`, `case-sensitive`, `--no-recursion`.

Example: `count-files --file-extension txt`

See also:

File searching by extension: list with file paths

2.11 File searching by pattern

You can also search for files using Unix shell-style wildcards: `*`, `?`, `[seq]`, `[!seq]` (with `-fm` or `--filename-match` argument). `*` - matches everything (zero or more occurrences of any character), `?` - matches any single character, `[seq]` - matches any character in `seq`, `[!seq]` - matches any character not in `seq`.

Some optional arguments are also available (`--file-sizes`, `--preview`, `--preview-size` arguments). If necessary, specify one or more of the common arguments: `path`, `--all`, `case-sensitive`, `--no-recursion`.

Example: `count-files --filename-match *.????`

See also:

File searching by pattern: list with file paths

2.12 Total counting of files

To count the total number of files, the number of files with a specific extension or the number of files without any extension you can use the `-t` or `--total` argument and specify the name of the extension.

You can also get a list of folders in which the found files are located, the number of found files in each folder and the total combined size of these files (`--show-folders` and `--total-size` optional arguments). When recursively counting all files (`--total .`) and using the `--show-folders` argument, all folders containing files are displayed.

If necessary, specify one or more of the common arguments: `path`, `--all`, `case-sensitive`, `--no-recursion`, `--no-feedback`.

Example: `count-files --total json`

See also:

Total number of files

2.13 Preview text files

Preview is available as an option when searching files using the `--filename-match` or `--file-extension` arguments.

The default text preview size depends on the terminal width settings. You can change this value by specifying the argument `-ps` or `--preview-size` followed by an integer (the number of characters to display from each file).

Example: `count-files --file-extension css --preview --preview-size 50`

2.14 File sizes

You can get additional information about the size of each file using the `-fs` or `--file-sizes` argument. This option is only available when searching files using the `--filename-match` or `--file-extension` arguments.

Example: `count-files --file-extension js --file-sizes`

When counting the total number of files (using `--total` argument) you can also get the total combined size of found files.

Example: `count-files --total py --total-size`

Examples of practical usage

More about the common arguments:

- `path` - *The path argument*
- `--all` - *Hidden files and directories*
- `--case-sensitive` - *Case sensitivity*
- `--no-recursion` - *Non-recursive search or counting*
- `--no-feedback` - *Customizing operation feedback*

3.1 File counting by extension: sorted table

Arguments in short form:

```
usage: count-files [-a] [-alpha] [-c]
                  [-nr] [-nf] [path]
```

Arguments in long form:

```
usage: count-files [--all] [--sort-alpha] [--case-sensitive]
                  [--no-recursion] [--no-feedback] [path]
```

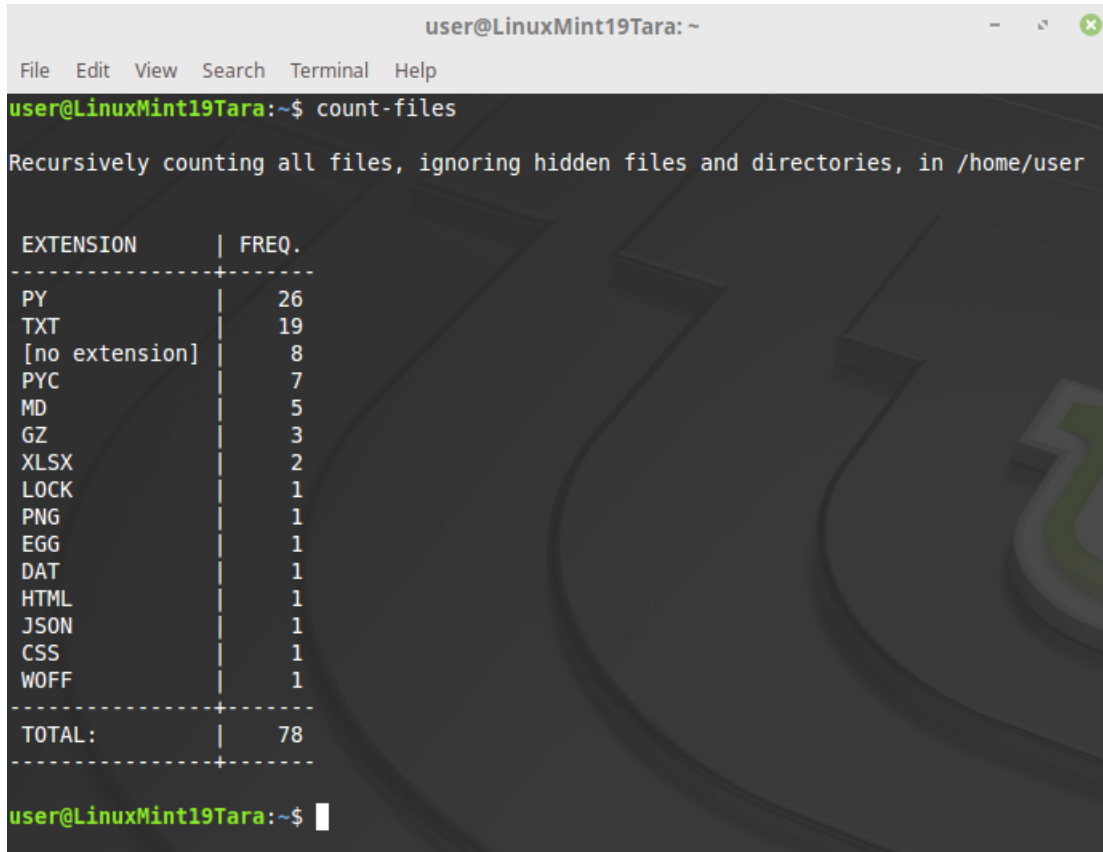
The most simple form of usage is to type a command in the shell, without any arguments.

By default, it will count files recursively in current working directory and all of its subdirectories, and will display a table showing the frequency for each file extension (e.g.: `.txt`, `.py`, `.html`, `.css`) and the total number of files found.

In this case, the file extensions in the table will be displayed in uppercase (default). Any hidden files or folders will be ignored.

Example:

```
count-files
```



```
user@LinuxMint19Tara: ~  
File Edit View Search Terminal Help  
user@LinuxMint19Tara:~$ count-files  
Recursively counting all files, ignoring hidden files and directories, in /home/user  


| EXTENSION      | FREQ. |
|----------------|-------|
| PY             | 26    |
| TXT            | 19    |
| [no extension] | 8     |
| PYC            | 7     |
| MD             | 5     |
| GZ             | 3     |
| XLSX           | 2     |
| LOCK           | 1     |
| PNG            | 1     |
| EGG            | 1     |
| DAT            | 1     |
| HTML           | 1     |
| JSON           | 1     |
| CSS            | 1     |
| WOFF           | 1     |
| TOTAL:         | 78    |

  
user@LinuxMint19Tara:~$
```

If you prefer alphabetically sorted results, you just need to add the `-alpha` or `--sort-alpha` argument.

Example with one of the common arguments:

```
count-files -c -alpha [path]  
count-files --case-sensitive --sort-alpha [path]
```



```

C:\Windows\system32\cmd.exe

Windows 8.1 > count-files --case-sensitive --sort-alpha E:\Photo

Recursively counting all files, ignoring hidden files and
directories, in E:\Photo

EXTENSION | FREQ.
-----+-----
docx      |    4
JPG       |   553
jpg       |  1129
zip       |    3
-----+-----
TOTAL:    |  1689
-----+-----

Windows 8.1 >

```

3.2 File searching by extension: list with file paths

Arguments in short form:

```
usage: count-files [-a] [-c] [-nr]
                  [-fe FILE_EXTENSION] [-fs]
                  [-p] [-ps PREVIEW_SIZE] [path]
```

Arguments in long form:

```
usage: count-files [--all] [--case-sensitive] [--no-recursion]
                  [--file-extension FILE_EXTENSION] [--file-sizes]
                  [--preview] [--preview-size PREVIEW_SIZE] [path]
```

This utility can also be used to search for files that have a certain file extension (using `-fe` or `--file-extension`) and, optionally, display a short preview (`-p` or `--preview`) for text files. The size of the preview text sample can optionally be customized by using the `-ps` or `--preview-size` argument followed by an integer number specifying the number of characters to present.

The list of file types for which preview is available can be viewed with the `-st` or `--supported-types` argument.

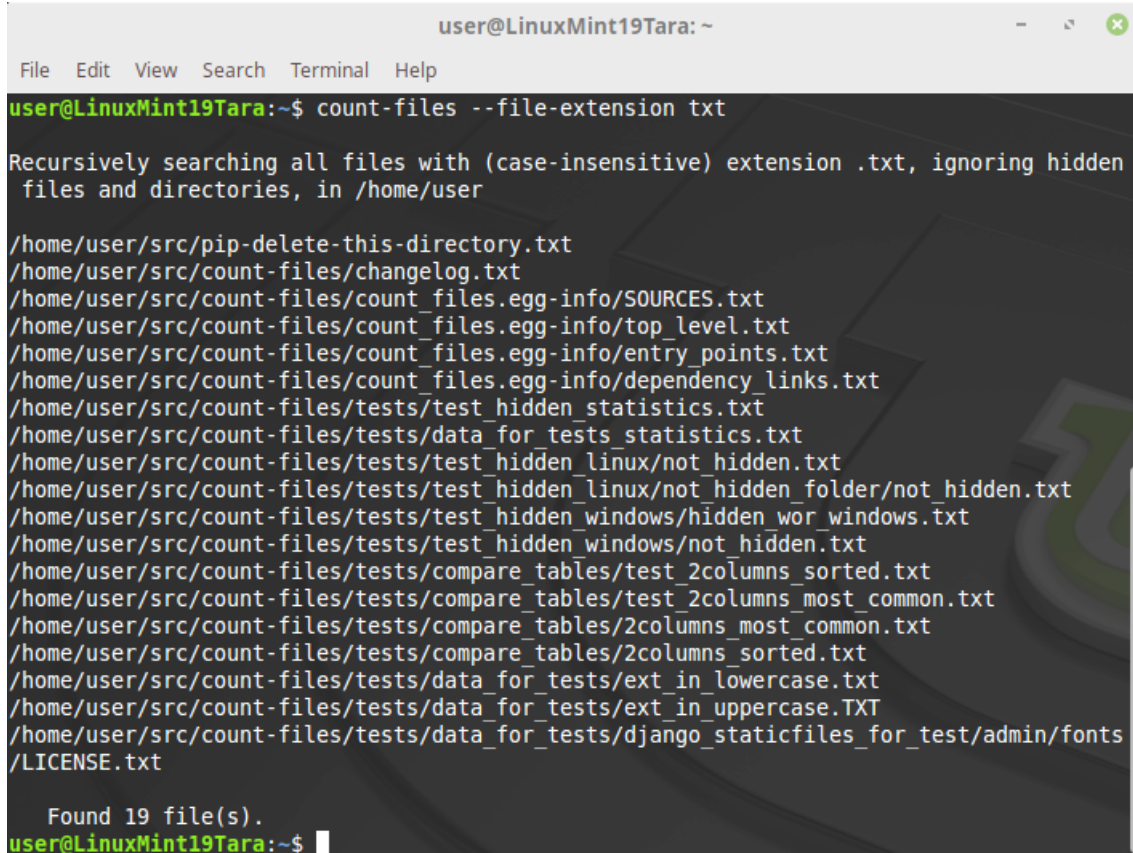
By default, the result of a search by a certain file extension is a list of the full paths of the files found. If you need information about the size of the files, use the `-fs` or `--file-sizes` argument.

3.2.1 Searching for files with a specific extension

Example:

```
count-files -fe txt

count-files --file-extension txt
```

A screenshot of a terminal window titled "user@LinuxMint19Tara: ~". The terminal shows the command "count-files --file-extension txt" being executed. The output lists 19 files with their full paths, all ending in ".txt". The files are located in various subdirectories under "/home/user/src/count-files". The terminal also shows the message "Found 19 file(s)." and the prompt "user@LinuxMint19Tara:~\$".

```
user@LinuxMint19Tara: ~
File Edit View Search Terminal Help
user@LinuxMint19Tara:~$ count-files --file-extension txt
Recursively searching all files with (case-insensitive) extension .txt, ignoring hidden
files and directories, in /home/user

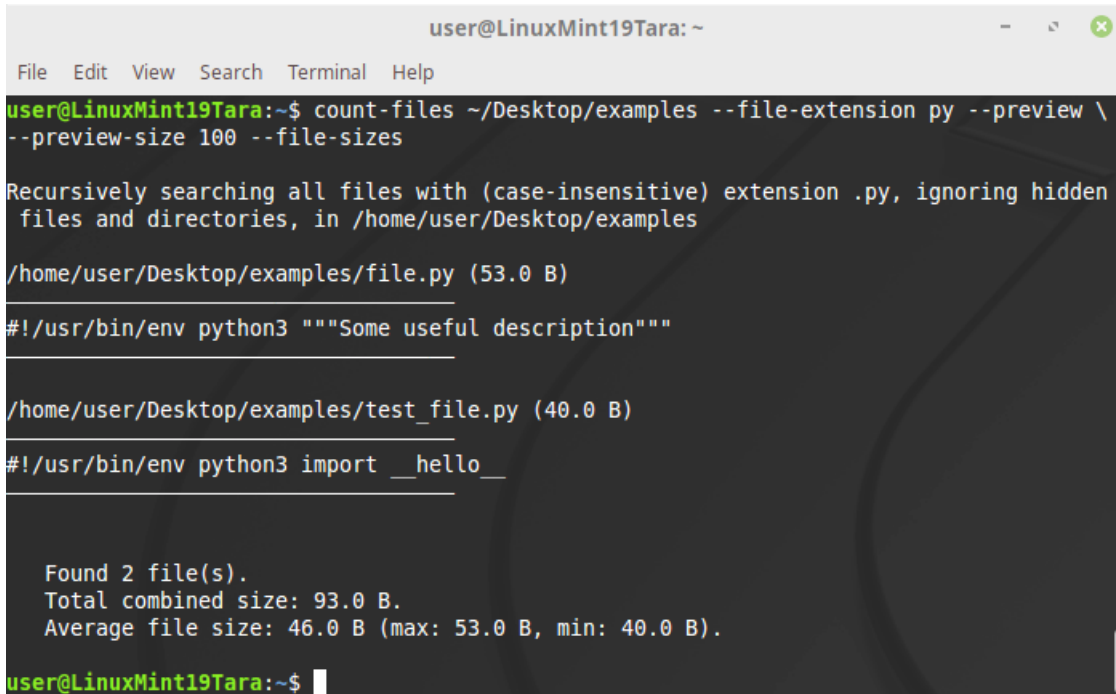
/home/user/src/pip-delete-this-directory.txt
/home/user/src/count-files/changelog.txt
/home/user/src/count-files/count_files.egg-info/SOURCES.txt
/home/user/src/count-files/count_files.egg-info/top_level.txt
/home/user/src/count-files/count_files.egg-info/entry_points.txt
/home/user/src/count-files/count_files.egg-info/dependency_links.txt
/home/user/src/count-files/tests/test_hidden_statistics.txt
/home/user/src/count-files/tests/data_for_tests_statistics.txt
/home/user/src/count-files/tests/test_hidden_linux/not_hidden.txt
/home/user/src/count-files/tests/test_hidden_linux/not_hidden_folder/not_hidden.txt
/home/user/src/count-files/tests/test_hidden_windows/hidden_wor_windows.txt
/home/user/src/count-files/tests/test_hidden_windows/not_hidden.txt
/home/user/src/count-files/tests/compare_tables/test_2columns_sorted.txt
/home/user/src/count-files/tests/compare_tables/test_2columns_most_common.txt
/home/user/src/count-files/tests/compare_tables/2columns_most_common.txt
/home/user/src/count-files/tests/compare_tables/2columns_sorted.txt
/home/user/src/count-files/tests/data_for_tests/ext_in_lowercase.txt
/home/user/src/count-files/tests/data_for_tests/ext_in_uppercase.TXT
/home/user/src/count-files/tests/data_for_tests/django_staticfiles_for_test/admin/fonts
/LICENSE.txt

Found 19 file(s).
user@LinuxMint19Tara:~$
```

Example:

```
count-files -fe py -p -ps 100 -fs [path]

count-files --file-extension py --preview
--preview-size 100 --file-sizes [path]
```



```

user@LinuxMint19Tara: ~
File Edit View Search Terminal Help
user@LinuxMint19Tara:~$ count-files ~/Desktop/examples --file-extension py --preview \
--preview-size 100 --file-sizes

Recursively searching all files with (case-insensitive) extension .py, ignoring hidden
files and directories, in /home/user/Desktop/examples

/home/user/Desktop/examples/file.py (53.0 B)
#!/usr/bin/env python3 """Some useful description"""

/home/user/Desktop/examples/test_file.py (40.0 B)
#!/usr/bin/env python3 import __hello__

Found 2 file(s).
Total combined size: 93.0 B.
Average file size: 46.0 B (max: 53.0 B, min: 40.0 B).
user@LinuxMint19Tara:~$

```

3.2.2 Searching and listing files without extension

Use a single dot `.` to search for files without any extension. Files with names such as `.gitignore`, `Procfile`, `_netrc` are considered to have no extension in their name.

Example: `count-files --file-extension . ~/Documents`

3.2.3 Searching and listing all files

Use two dots without spaces `..` to search for all files with or without file extensions in their names.

Example: `count-files --file-extension .. ~/Documents`

3.3 File searching by pattern: list with file paths

Arguments in short form:

```
usage: count-files [-a] [-c] [-nr]
                  [-fm PATTERN] [-fs]
                  [-p] [-ps PREVIEW_SIZE] [path]
```

Arguments in long form:

```
usage: count-files [--all] [--case-sensitive] [--no-recursion]
                [--filename-match PATTERN] [--file-sizes]
                [--preview] [--preview-size PREVIEW_SIZE] [path]
```

Search for files using Unix shell-style wildcards: `*`, `?`, `[seq]`, `[!seq]`.

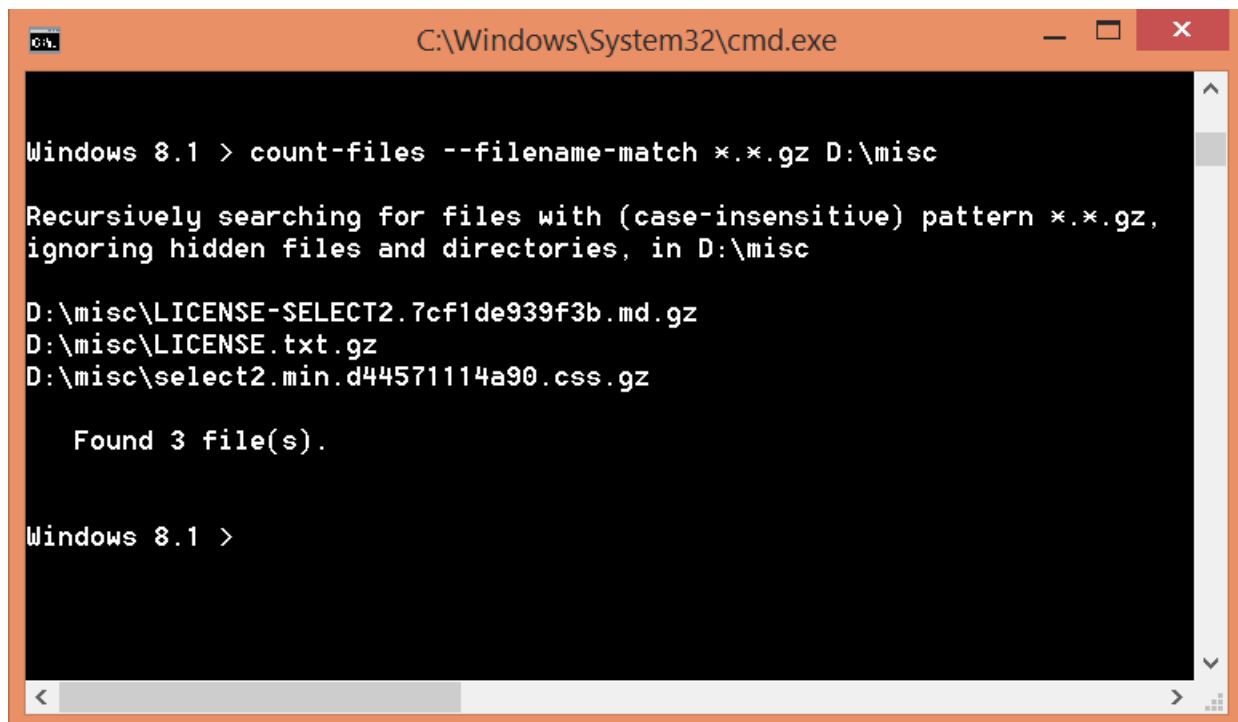
`*` - matches everything (zero or more occurrences of any character), `?` - matches any single character, `[seq]` - matches any character in `seq`, `[!seq]` - matches any character not in `seq`. For a literal match, wrap the meta-characters in brackets. For example, `"[?]"` matches the character `"?"`.

Optionally, you can get a short preview for text files (`-p` or `--preview` argument). The size of the preview text sample can be customized by using the `-ps` or `--preview-size` argument. The list of file types for which preview is available can be viewed with the `-st` or `--supported-types` argument. If the pattern matches both supported and unsupported file extensions, then a preview of unsupported files will be skipped.

If you need information about the size of the files, use the `-fs` or `--file-sizes` argument.

Example:

```
count-files -fm *.*.gz [path]
count-files --filename-match *.*.gz [path]
```

A screenshot of a Windows 8.1 command prompt window titled "C:\Windows\System32\cmd.exe". The window has a black background with white text. The user has entered the command "count-files --filename-match *.*.gz D:\misc". The output shows a recursive search for files with the pattern *.*.gz in the directory D:\misc, listing three files: "D:\misc\LICENSE-SELECT2.7cf1de939f3b.md.gz", "D:\misc\LICENSE.txt.gz", and "D:\misc\select2.min.d44571114a90.css.gz". It concludes with "Found 3 file(s)." and returns to the prompt "Windows 8.1 >".

```
C:\Windows\System32\cmd.exe

Windows 8.1 > count-files --filename-match *.*.gz D:\misc

Recursively searching for files with (case-insensitive) pattern *.*.gz,
ignoring hidden files and directories, in D:\misc

D:\misc\LICENSE-SELECT2.7cf1de939f3b.md.gz
D:\misc\LICENSE.txt.gz
D:\misc\select2.min.d44571114a90.css.gz

    Found 3 file(s).

Windows 8.1 >
```

3.3.1 Other examples with wildcards

Example for `.pyc`, `.pyo`, `.pyd` and similar files:

```
count-files --filename-match *.py? [path]
```

Example for file names containing the word “test”:

```
count-files --filename-match *test* [path]
```

Example for file names starting with the word “LICENSE” in uppercase:

```
count-files --filename-match LICENSE* --case-sensitive [path]
```

Example for exact match of the file name:

```
count-files --filename-match setup.py [path]
```

3.4 Total number of files

Arguments in short form:

```
usage: count-files [-a] [-c] [-nr] [-nf] [-t EXTENSION]
                  [-sf] [-ts] [path]
```

Arguments in long form:

```
usage: count-files [--all] [--case-sensitive] [--no-recursion]
                  [--no-feedback] [--total EXTENSION]
                  [--show-folders] [--total-size] [path]
```

If you only need the total number of all files, or the number of files with a certain extension or without it, use the `-t` or `--total` argument.

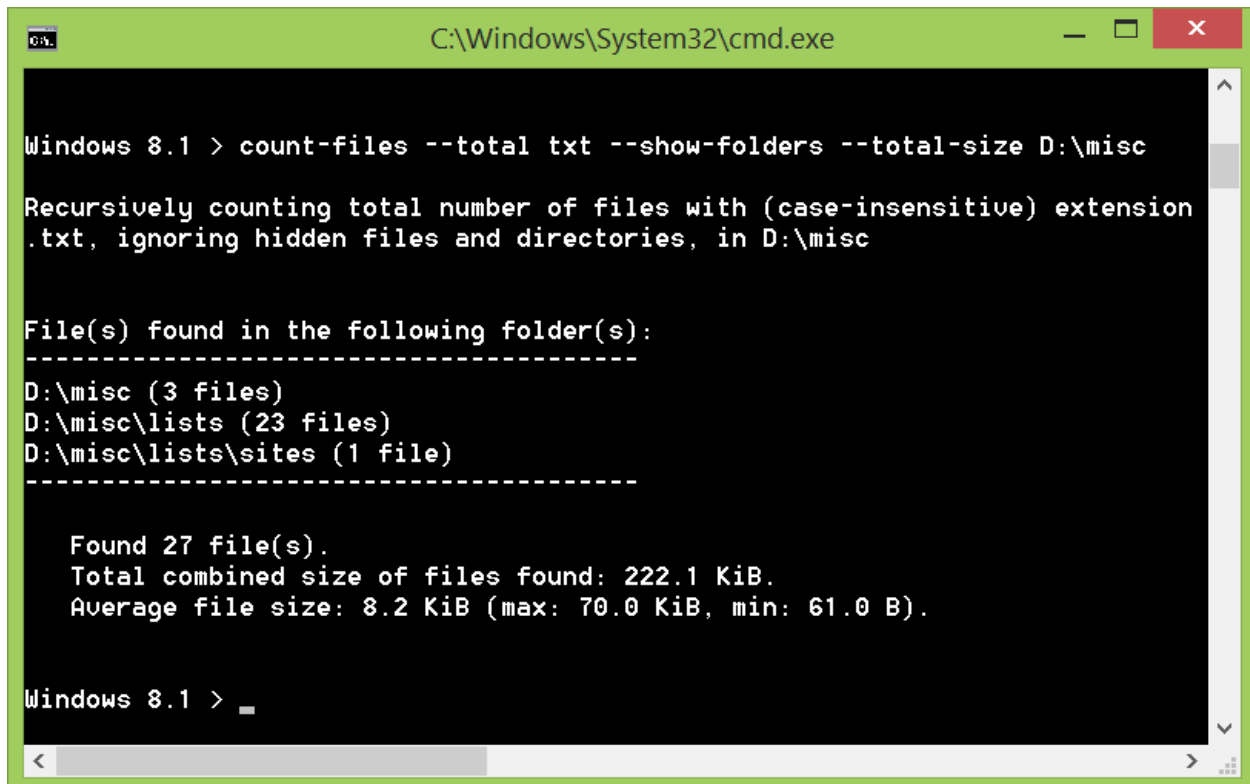
To count the total number of files, you must specify the name of the extension.

3.4.1 Total number of files: counting files with a specific extension

Example:

```
count-files -t txt -sf -ts [path]

count-files --total txt --show-folders --total-size [path]
```

A screenshot of a Windows 8.1 command prompt window titled "C:\Windows\System32\cmd.exe". The window has a green title bar and standard Windows window controls. The command prompt shows the following text:

```
Windows 8.1 > count-files --total txt --show-folders --total-size D:\misc

Recursively counting total number of files with (case-insensitive) extension
.txt, ignoring hidden files and directories, in D:\misc

File(s) found in the following folder(s):
-----
D:\misc (3 files)
D:\misc\lists (23 files)
D:\misc\lists\sites (1 file)
-----

Found 27 file(s).
Total combined size of files found: 222.1 KiB.
Average file size: 8.2 KiB (max: 70.0 KiB, min: 61.0 B).

Windows 8.1 > _
```

3.4.2 Total number of files: counting files without extension

Use a single dot `.` to get the total number of files that don't have an extension.

Example: `count-files --total . ~/Documents`

3.4.3 Total number of files: counting all files

Use two dots without spaces `..` to get the total number of files, with or without a file extension.

Example: `count-files --total .. ~/Documents`

Did you find a bug or do you have a suggestion?

Please, open a new issue or a pull request to the [repository](#).

CHAPTER 4

Search

- search