

## 42 -Finding files - find

**find** : Short examples: Searches recursively for files or directories through a directory tree.

### Syntax:

```
find startdirectory [search_criteria_options] [-exec command \;]
find startdirectory [search_criteria_options] [-ok command \;]
                                     (ok asks to confirm)
```

### Examples:

```
find Path -name Datei    (Name should always be inside " " when * is used)
find Path -regex "pattern"
                               Uses a regular expression to find the filename
find Path -type f         (f= regular file, d = Directory, l = Symlink)
find Path -user Benutzername
find Path -maxdepth Tiefe
find Path -xdev            (doesn't search inside mounted filesystems)
find Path -cmin -Minuten
find Path -follow         (follow symlinks)
find Path -perm +u+s      Finds all files having the SUID turned on.
find Path -size +xx{c|k|M|G} Finding file with size greater than
                               xx {c bytes | kilobytes | Megabytes | Gigabytes}
```

### More Examples:

```
find . type d -maxdepth 1 | sort
```

Finds all directories located in the current directory and sort them

```
find / -perm -u+s
```

Finds all the files that have the SUID bit set in system.

```
cd /etc/ ; find . -name "*XF*"
```

Finds all files recursively in /etc directory of which their names includes the pattern 'XF'

```
find /opt/kde3 -maxdepth 2 -type f -name "*edit"
```

Searches in /opt/kde3 and in 2 subdirectories deep, any file of which their name ends-up with the word 'edit'

```
find . -follow -cmin -5
```

Search the files that of which their properties have been changed less than 5 minutes ago)

-cmin +5 Properties of file changed more than 5 minutes ago

-amin -6 Content of file accessed less than 6 minutes ago

-mmin +8 Content of file modified more than 8 minutes ago

-ctime +5 Properties of file changed more than 5 days ago

-atime -7 Content of file accessed less than 7 days ago

-mtime -3 Content of file modified less than 3 days ago

```
find /etc -type f -name '*.conf' -exec grep -H "hosts" {} \;
```

Executes the grep on each found file. Each found line will be shown accompanied with the name of the file where it was found.

```
find /etc -type f -name '*.conf' -ok grep -H "hosts" {} \;
```

Same actions as above except that -ok option asks find to prompt for confirmation (with **y** or **n**) of the command to do before executing it.

**locate** : Locate files in the whole system based using a database of filenames.

Syntax:

```
locate filename
```

Searched in the locate database for the *filename*.  
This database is in `/var/lib/locatedb`  
It is updated via the command: `updatedb [options]`  
The configuration file for `updatedb` is `/etc/updatedb.conf`

`-d path, --database=path`  
Instead of searching the default file name database, search the file name databases in `path`, which is a colon-separated list of database file names. You can also use the environment variable `LOCATE_PATH` to set the list of database files to search. The option overrides the environment variable if both are used.

`-e, --existing`  
Only print out such names that currently exist (instead of such names that existed when the database was created). Note that this may slow down the program a lot, if there are many matches in the database.

`-i, --ignore-case` Ignore case distinctions in both the pattern and the file names.

- **slocate** : Secure locate of file in system

Syntax:

```
slocate [options] filename
```

Secure Locate provides the same features as the `locate` but it will also store file permissions and ownership so that users will not see files they do not have access to. `slocate` database is not the same as `locate` database.

It needs to be built by issuing the `slocate` command with proper options:

Database Build Options:

<code>-u</code>	Create <code>slocate</code> database starting at root directory <code>'/ '</code>
<code>-U &lt;dir&gt;</code>	Create <code>slocate</code> database starting at path <code>&lt;dir&gt;</code>
<code>-e &lt;dir1,dir2,...&gt;</code>	Exclude directories from <code>slocate</code> database .
<code>-f &lt;fstype1,...&gt;</code>	Exclude files on specific file systems from the <code>slocate</code> database.
<code>-c</code>	Parse <code>/etc/updatedb.conf</code> when updating the <code>slocate</code> database.
<code>-l &lt;level&gt;</code>	Security level:

	0	Turns security checks off. This will make search faster.
	1	Turns security checks on. This is the default.
<code>-o &lt;file&gt;</code>		Specifies the database to create.
<code>--output=&lt;file&gt;</code>		
<code>-v, --verbose</code>		Verbose mode. Display files when creating database.

Slocate Search Options:

<code>-i</code>	Does a case insensitive search.
<code>-q</code>	Quiet mode. Error messages are suppressed.
<code>-n &lt;num&gt;</code>	Limit the amount of results shown to <num>.
<code>--regexp=&lt;regexp&gt;</code>	
<code>-r &lt;regexp&gt;</code>	Search the database using a basic POSIX regular expression.
<code>-d &lt;path&gt;</code>	Specifies the path of databases to search in. <code>--database=&lt;path&gt;</code>

- **whereis**: Search for a program and possibly its man pages from a predefined path.

Syntax:

`whereis filename` Searches a predefined (hard coded) list of directories for the filename and man pages. They must be in the path predefined during compilation of `whereis` program.

- **which** : Search for the first occurrence of a program through the `PATH`.

Syntax:

`which filename` Searches the `PATH` for the first occurrence of the filename. The filename can be a list of files.

`type -p filename` Same as above `which filename`