

How to search text strings only in hidden files dot (.) files within a directory on Linux and FreeBSD

Author : admin



If there is necessity to look for a string in all hidden files with all sub-level subdirectories (be aware this will be time consuming and CPU stressing) use:

```
hipo@noah:~$ grep -rli 'PATH' .*
```

```
./gftp/gftp.rc  
./gftp/cache/cache.OOqZVP  
....
```

Sometimes its necessary to only grep for variables within the first-level directories (lets say you would like to *grep* a 'PATH' variable set, string within the \$HOME directory, the command is:

```
hipo@noah:~$ grep PATH .[!]*
```

```
.profile:PATH=/bin:/usr/bin/:${PATH}  
.profile:export PATH
```

```
.profile:# set PATH so it includes user's private bin if it exists
.profile: PATH="$HOME/bin:$PATH"
.profile.language-env-bak:# set PATH so it includes user's private bin if it exists
.profile.language-env-bak: PATH="$HOME/bin:$PATH"
.viminfo:~/PATH.xcyrillic: XNLSPATH=/usr/X11R6/lib/X11/nls
.xcyrillic: export XNLSPATH
```

The regular expression `.[!]*`, means exclude any file or directory name starting with '.', e.g. match only `.*` files

Note that to use the `grep PATH .[!]*` on FreeBSD you will have to use this regular expression in bash shell, the default BSD `cs`h or `ts`ch shells will not recognize the regular expression, e.g.:

```
grep PATH '[!]*'
grep: .[!]*: No such file or directory
```

Hence on BSD, if you need to look up for a string within the home directory, hidden files: `.profile .bashrc .bash_profile .cshrc` run it under bash shell:

```
freebsd# /usr/local/bin/bash
[root@freebsd:/home/hipo]# grep PATH .[!]*

.bash_profile:# set PATH so it includes user's private bin if it exists
.bash_profile:# PATH=~/.bin:${PATH}
.bash_profile:# do the same with ...
```

Another easier to remember, alternative `grep` cmd is:

```
hipo@noah:~$ grep PATH .*
.profile:PATH=/bin:/usr/bin:${PATH}
```

```
.profile:export PATH
.profile:# set PATH so it includes user's private bin if it exists
.profile: PATH="$HOME/bin:$PATH"
....
```

Note that *grep 'string' .** is a bit different in meaning, as it will not prevent *grep* to match filenames with names *..filename1*, *..filename2* etc.

Though *grep 'string' .** will work note that it will sometimes output some unwanted matches if filenames with double dot in the beginning of file name are there ...

That's all folks :)