

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:



Ask programming questions



Answer and help your peers

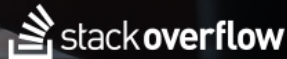


Get recognized for your expertise

bash echo number of lines of file given in a bash variable without the file name

Work on work you love.

From home.



I have the following three constructs in a bash script:

```
NUMOFLINES=$(wc -l $JAVA_TAGS_FILE)
echo $NUMOFLINES" lines"

echo $(wc -l $JAVA_TAGS_FILE)" lines"

echo "$(wc -l $JAVA_TAGS_FILE) lines"
```

And they both produce identical output when the script is run:

```
121711 /home/slash/.java_base.tag lines
121711 /home/slash/.java_base.tag lines
121711 /home/slash/.java_base.tag lines
```

I.e. the name of the file is also echoed (which I don't want to). Why do these scriptlets fail and how should I output a clean:

```
121711 lines
```

?

bash wc

edited Nov 19 '15 at 23:04



Ciro Santilli 巴拿馬文件
六四事件 法輪功
42.3k ● 8 ● 183 ● 141

asked Aug 18 '12 at 21:15



Marcus Junius Brutus
6,794 ● 7 ● 62 ● 122

Possible duplicate of [get just the integer from wc in bash](#) – Ciro Santilli 巴拿馬文件 六四事件 法輪功 Nov 19 '15 at 23:05

6 Answers

An Example Using Your Own Data

You can avoid having your filename embedded in the **NUMOFLINES** variable by using redirection from **JAVA_TAGS_FILE**, rather than passing the filename as an argument to **wc**. For example:

```
NUMOFLINES=$(wc -l < "$JAVA_TAGS_FILE")
```

Explanation: Use Pipes or Redirection to Avoid Filenames in Output

The **wc** utility will not print the name of the file in its output if input is taken from a pipe or redirection operator. Consider these various examples:

```
# wc shows filename when the file is an argument
$ wc -l /etc/passwd
41 /etc/passwd

# filename is ignored when piped in on standard input
$ cat /etc/passwd | wc -l
41

# unusual redirection, but wc still ignores the filename
$ < /etc/passwd wc -l
41

# typical redirection, taking standard input from a file
$ wc -l < /etc/passwd
41
```

As you can see, the only time **wc** will print the filename is when its passed as an argument, rather than as data on standard input. In some cases, you may want the filename to be printed, so it's useful to understand when it will be displayed.

answered Aug 18 '12 at 21:31



CodeGnome
39k ● 4 ● 57 ● 114

beware that this approach will not include the last line if the last line does not terminate by end of line character. See my fix below. – ling Oct 18 '15 at 5:33



Did you find this question interesting? Try our newsletter

Sign up for our newsletter and get our top new questions delivered to your inbox ([see an example](#)).

wc can't get the filename if you don't give it one.

```
wc -l < "$JAVA_TAGS_FILE"
```

answered Aug 18 '12 at 21:17

[Ignacio Vazquez-Abrams](#)

429k ● 59 ● 760 ● 924

- 2 Well, then don't. Just pass the file to wc's stdin as shown in the answer. So `cat "$JAVA_TAGS_FILE" | wc -l` or, equivalently, `wc -l < "$JAVA_TAGS_FILE"`. This way, wc only gets raw data, not the filename. – [Witiko](#) Jul 4 '13 at 14:23

You can also use awk:

```
awk 'END {print NR,"lines"}'
filename
```

Or

```
awk 'END {print NR}' filename
```

answered Aug 11 '15 at 12:33

[chilicuil](#)

231 ● 2 ● 9

It's a very simple:

```
NUMOFLINES=$(cat $JAVA_TAGS_FILE | wc -l )
```

or

```
NUMOFLINES=$(wc -l $JAVA_TAGS_FILE | awk '{print $1}')
```

answered Aug 18 '12 at 21:18

[Slava Semushin](#)

8,259 ● 5 ● 29 ● 44

(apply on Mac, and probably other Unixes)

Actually there is a problem with the wc approach: it does not count the last line if it does not terminate with the end of line symbol.

Use this instead

```
nblines=$(cat -n file.txt | tail -n 1 | cut -f1 | xargs)
```

or even better (thanks gniourf_gniourf):

```
nblines=$(grep -c '' file.txt)
```

Note: The awk approach by chilicuil also works.

edited Oct 19 '15 at 7:41

answered Oct 18 '15 at 5:32

[ling](#)

1,208 ● 1 ● 12 ● 18

- 1 Very convoluted method! Maybe you'll want `nblines=$(grep -c '' file)` instead (which is the canonical way of counting incomplete lines in this case). Note though that according to POSIX, you're counting the [incomplete lines](#) (and not the [lines](#)). You're in fact dealing with a binary file and not a [text file](#). – [gniourf_gniourf](#) Oct 18 '15 at 6:33

@gniourf_gniourf Thanks, I didn't know about that, it works great and is even more concise. – [ling](#) Oct 19 '15 at 7:39

I normally use the 'back tick' feature of bash

```
export NUM_LINES=`wc -l filename`
```

Note the 'tick' is the 'back tick' e.g. ` not the normal single quote

edited Jan 1 '15 at 20:03

answered Jul 18 '14 at 12:45

[fejese](#)

3,398 ● 4 ● 16 ● 29

[Russ Hore](#)

1

- 5 That's just a different notation, and doesn't solve the issue of the file name being part of the result. – [lizzy](#) Jan 1 '15 at 19:57