

## Re: [SLE] bash Script Using Filenames with Embedded Spaces

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/SuSE/2006-11/msg00964.html>

---

- *From:* Randall R Schulz <[rschulz@xxxxxxxxx](mailto:rschulz@xxxxxxxxx)>
  - *Date:* Tue, 7 Nov 2006 07:46:44 -0800
- 

On Monday 06 November 2006 23:55, Darryl Gregorash wrote:

...

If the code fragment you gave above works, then what appears to be spaces are some other character code, not an ASCII space (040 / 0x20). There are other character codes that make no mark and are not zero-width. Perhaps your file names use one of them.

That simply is not possible. The list is determined by "for i in <list>" and that list is determined prior to the actual parsing of any commands inside the for loop. If it is a literal space for the ls command, then it must be a literal space for any other command. But the variation of results reported here, and your comments in particular, have made me wonder what effect the quotes will have placed in different places inside the script.

In this fragment:

```
% for VAR_NAME in GLOB ...
```

The glob is expanded by the for loop processing and is not subject to further argument list parsing for the purpose of producing successive values to assign to VAR\_NAME, so spaces in the names don't confuse the iteration. There is always exactly one iteration for each file name matched by GLOB.

Nonetheless, my point was that if you have a character such as an en-space (or it's fatter sibling, em-space, which is the width of a lower-case 'm'), it can be expanded unquoted in an argument list without the en-space causing a break in the argument as a regular space would.

## Re: [SLE] bash Script Using Filenames with Embedded Spaces

First I changed my test script to execute a "ls -l \$i" after the echo, and came up with this sort of result:

```
Antonio Vivaldi-Gloria Dixit Dominus Magnificat Choir of Kings  
College Cambridge and the Academy of Ancient Music.m3u  
ls: Antonio: No such file or directory  
ls: Vivaldi-Gloria: No such file or directory  
ls: Dixit: No such file or directory  
ls: Dominus: No such file or directory  
(etc)
```

exactly as you stated.

The best I can think of is that the ls command takes a list, ie. the actual ls commandline format is not "ls [options] [filespec]", rather it is "ls [options] [list\_of\_filespecs]". Any command behaving in the same way, such as cp, will fail in the same way. When \$i in the ls command is placed inside double quotes, I get the expected ls output:

Absolutely not. Ls is not special in this way. The only special treatment is the expansion of the glob pattern and the assignment of the resulting file names to the for-loop variable without further argument parsing occurring. Once the for-loop variable is referenced in the body of the loop, quoting again becomes significant and an unquoted reference when the value contains shell-significant syntactic characters (space, [, \*, \*, ? | and so on) will yield to unexpected and undesired results.

```
~>cat thingy  
cd /pub/music/mp3  
for i in *.m3u; do echo $i; ls -l "$i"; done  
cd ~  
~>./thingy  
Antonio Vivaldi-Gloria Dixit Dominus Magnificat Choir of Kings  
College Cambridge and the Academy of Ancient Music.m3u  
-rw-r--r-- 1 raven users 5364 2005-11-05 01:31 Antonio  
Vivaldi-Gloria Dixit Dominus Magnificat Choir of Kings College  
Cambridge and the Academy of Ancient Music.m3u  
(etc)
```

If the double quotes are replaced with single quotes, each ls command returns an error, "ls: \$i: No such file or directory". I've forgotten the subtle nuances of quotes in the bash commandline to remember why this happens.

Within double-quotes variable and command-to-argument expansion (back-ticks or \$( ... )) are expanded. Within single quotes \_nothing\_

## Re: [SLE] bash Script Using Filenames with Embedded Spaces

is expanded.

When the for loop is written the following way (and these can be single or double quotes, it does not matter), then there is a single variable in the list of the for loop, so the echo and ls commands are actually run just once each:

```
for i in '*.m3u'; do echo $i; ls -l $i; done
```

and the output becomes:

```
~> ./thingy
Antonio Vivaldi-Gloria Dixit Dominus Magnificat Choir of Kings
College Cambridge and the Academy of Ancient Music.m3u Georg
...
-rw-r--r-- 1 raven users 5364 2005-11-05 01:31 Antonio
Vivaldi-Gloria Dixit Dominus Magnificat Choir of Kings College
Cambridge and the Academy of Ancient Music.m3u
-rw-r--r-- 1 raven users 624 2005-11-19 03:12 Georg Friederich
Händel-Water Music Suite.m3u
(etc)
```

In this case, the glob expansion of \*.m3u is happening upon each unquoted reference in the body of the loop.

**Lesson: ALWAYS QUOTE VARIABLE REFERENCES WHOSE  
CONTENT YOU DO NOT  
FULLY CONTROL.**

So, that is absolutely correct, but with the qualification that what you get does depend on where and how you do the quoting, and AFAICT there is no hard and fast rule that gives a single correct way to use quotes.

No, there's not. One cannot write non-trivial programs in any programming language blindly, based only on rules-of-thumb without really knowing that language and whatever subtleties it may have. BASH has a few...

Did someone say the bash commandline isn't interesting and fun? :-)

Re: [SLE] bash Script Using Filenames with Embedded Spaces

I think so. I do tons of BASH programming.

Randall Schulz

—

Check the headers for your unsubscribe address

For additional commands send e-mail to [suse-linux-e-help@xxxxxxxx](mailto:suse-linux-e-help@xxxxxxxx)

Also check the archives at <http://lists.suse.com>

Please read the FAQs: [suse-linux-e-faq@xxxxxxxx](mailto:suse-linux-e-faq@xxxxxxxx)