

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

Source: <http://linux.derkeiler.com/Mailing-Lists/SuSE/2006-11/msg00923.html>

- *From:* Darryl Gregorash <raven@xxxxxxxxxxxxxx>
 - *Date:* Tue, 07 Nov 2006 01:55:30 -0600
-

On 2006-11-07 00:06, Randall R Schulz wrote:

On Monday 06 November 2006 21:11, Lucky Leavell wrote:

...

OK, here is my complete script which runs under bash on an Ubuntu 6.06 system where I have the mp32ogg utility to convert mp3 to ogg files which k3b can handle:

```
for i in *.mp3
do
echo $i
mp32ogg $i
done
```

which works fine if the only craziness is embedded spaces in the file name; it failed when there were parentheses but, since that is rare in my situation, I can live with it. (Of course, if the single quoted \$i would work there ... I'm off to try it!)

<snip>

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

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

wonder what effect the quotes will have placed in different places inside the script.

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 in