

Re: problems with opening files due to file's path

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-06/msg01255.html>

- *From:* Lie <Lie.1296@xxxxxxxxxx>
 - *Date:* Wed, 11 Jun 2008 06:48:45 -0700 (PDT)
-

On Jun 11, 10:07 am, Alexnb <alexnbr...@xxxxxxxxxx> wrote:

I don't think you understand it doesn't matter how the variable gets there, the same code is run regardless, I have no problem with the GUI, but you asked, and so I told you. the code `os.startfile(...)` is run if there is a GUI or it is a console app.

(snip)

I think I know why you get confused by this, clearly, you have no idea of what an escape character and escape sequence is.

Python (and C/C++ and some other languages) treats `\` (the backspace character) inside a string specially, it makes the character after the backspace lose their special meaning OR get a special meaning, you might probably be used to seeing something like this: 'First line `\n`Second Line', which when printed, would give:

```
print 'First Line\nSecond Line'
```

First Line
Second Line

The second behavior of the escape sequence is to make special character (generally the backspace itself), lose their special meaning:

```
print 'path\\file.txt'
```

path\file.txt

In some cases, you might sometimes want a path like this: 'path `\nick.txt`' if you do this:

```
print 'path\nick.txt'
```

Re: problems with opening files due to file's path

```
path  
ick.txt
```

because the `\n` is considered as a newline.

Instead, you should do this:

```
print 'path\\nick.txt'
```

```
path\nick.txt
```

or

```
print r'path\nick.txt'
```

```
path\nick.txt
```

the `r` string is raw string, most of the magics of a regular string " is lost for an `r` string. It allows you to avoid the need to escape the special characters. Raw string is usually used for regular expressions and paths in Windows both of which uses the backslash character regularly.

you first case of:

```
system("C:\Documents and Settings\Alex\My Documents\My Music\Rhapsody  
\Bryanbros\Weezer\2001\04 – Island In The Sun.wma")
```

is interpreted by python as this:

```
"C:\Documents and Settings\Alex\My Documents\My Music\Rhapsody  
\Bryanbros\Weezer\2001 – Island In The Sun.wma"
```

Notice that the `\0` is substituted into `' '`, because `\0` is the escape sequence for null character.

(Personally I think python should raise errors if any escape character that isn't followed by a valid escape sequence is found (such as `\D`, `\A`, `\M`, `\R`, `\B`, `\W`, `\()`, but perhaps they're trying not to be too mean for newbies.)

that should be correctly written like this:

```
system(r"C:\Documents and Settings\Alex\My Documents\My Music\Rhapsody  
\Bryanbros\Weezer\2001\04 – Island In The Sun.wma")
```

or:

```
system("C:\\Documents and Settings\\Alex\\My Documents\\My Music\\  
Rhapsody\\Bryanbros\\Weezer\\2001\\04 – Island In The Sun.wma")
```

Now, to the next question:

How if I want the path to come from a variable:

```
path = "C:\Documents and Settings\Alex\My Documents\My Music\Rhapsody  
\Bryanbros\Jason Mraz\I'm Yours (Single)\01 – I'm Yours.wma"  
os.startfile(path)
```

Re: problems with opening files due to file's path

I can see in a glance why it doesn't work, the string is escaped by python, it should be written like this.

```
path = r"C:\Documents and Settings\Alex\My Documents\My Music\Rhapsody  
\Bryanbros\Jason Mraz\I'm Yours (Single)\01 - I'm Yours.wma"  
os.startfile(path)
```

OR:

```
path = "C:\\Documents and Settings\\Alex\\My Documents\\My Music\\  
Rhapsody\\Bryanbros\\Jason Mraz\\I'm Yours (Single)\\01 - I'm  
Yours.wma"  
os.startfile(path)
```

In most GUI toolkits (including Tkinter) and `raw_input()` function, when you input a string (using the textbox, a.k.a Entry widget) it would automatically be escaped for you, so when you input 'path\path\file.txt', the GUI toolkit would convert it into 'path\\path\\file.txt'.

.