

Don't forget to pass the current directory along with the command line to your single-instance program

 devblogs.microsoft.com/oldnewthing/20070629-00

June 29, 2007



Raymond Chen

If you make a single-instance program, and somebody runs a second copy of the program and passes a command line, the most common way of handling this is to hand the command line to the first copy of the program and let the first copy deal with it. When you do this, don't forget about the current directory.

If somebody passes a relative path to the second copy of the program, that relative path needs to be resolved against the current directory of the *second* program. I've seen programs that fail to take this into account. Instead, they pass the command line to the first copy of the program, and the first copy resolves the relative paths against *its* current directory.

Allow me to give a concrete example.

```
C:\Directory1> start LitWare file1.lit
... runs LitWare with the file C:\Directory1\file1.lit ...
C:\Directory1> cd ..\Directory2
C:\Directory2> start LitWare file2.lit
```

What you expect to happen is that LitWare opens the file `C:\Directory2\file2.lit`, since the relative path `file2.lit` should be resolved against the current directory, `C:\Directory2`. Unfortunately, I see some programs try to open the file `C:\Directory1\file2.lit` since they passed the command line to the first copy, and the first copy then resolved the relative path `file2.lit` against the current directory of the first copy, namely `C:\Directory1`.

Result: "File not found error."

Moral of the story: Be mindful of the current directory when parsing the command line. You can either have the second copy parse the command line (and resolve the relative paths against its own current directory), or you can pass the current directory to the first copy (and resolve the relative paths against that directory). Either works. What doesn't work is passing the relative paths to the first copy and having the first copy resolve it against its own current directory.

My workaround for dealing with these programs is using the `%CD%` pseudo-variable.

```
C:\Directory2> start LitWare %CD%\file2.lit
```

The `%CD%` pseudo-variable expands to the command prompt's current directory. (Don't forget to enclose it in quotation marks if the path contains a space.)

Nitpicker's corner

◇Remember that the phrase “some programs” means “some programs” and Microsoft programs fall under the category of “programs”.☺

☺Yes, I'm using strange symbols as note markers. My hope is that the people who nitpick over my choice of note markers will get so upset that their heads will explode and they therefore will be unable to nitpick further. It appears that some people, on the other hand, suffer from some medical condition that disables the “fun” center of the brain.

[Update: Give IE6 a little font help.]

Raymond Chen

Follow

