

The redirection can come anywhere on the line, so watch out for those spaces

 devblogs.microsoft.com/oldnewthing/20170731-00

July 31, 2017



Raymond Chen

Welcome to Batch File Week. Batch is probably one of the most hated programming languages that people are still writing new programs in. You hate it, but you have to deal with it. So just deal with it.

That said, here we go.

A customer couldn't understand why parenthesizing a few `ECHO` statements resulted in a syntax error.

This version of the batch file works:

```
@echo off
set log=C:\Space Path

echo Test >>%log%\Output.txt
echo Redirection >>%log%\Output.txt
```

On the other hand, this version, which should be equivalent, spits out a syntax error.

```
@echo off
set log=C:\Space Path

(
    echo Test
    echo Redirection
) >>%log%\Output.txt
```

The error is

```
Path\Output.txt was unexpected at this time.
```

If the syntax of the second command is illegal, then so too should the first, since they are basically the same thing. Why is one legal and the other not?

Recall that you're allowed to put the redirection operator anywhere on the line, and when it is parsed, it is removed from the command line, and what remains needs to be a legal command.

In the first batch file, the `echo` statement expands to

```
echo Test >>C:\Space Path\Output.txt
```

The redirection operator detector takes out the redirection operator and the file name (which due to the embedded space is parsed as merely `C:\Space`), leaving

```
echo Test Path\Output.txt
```

Result: The string `Test Path\Output.txt` is placed into the file `C:\Space`.

Now let's look at the second batch file. The command being parsed is

```
(
  echo Test
  echo Redirection
) >>%log%\Output.txt
```

which expands to

```
(
  echo Test
  echo Redirection
) >>C:\Space Path\Output.txt
```

Again, the redirection operator detector takes out the redirection operator and the file name, leaving

```
(
  echo Test
  echo Redirection
) Path\Output.txt
```

And that is a syntax error.

In other words, the first batch file works because the extra junk you appended happened to be in a place where junk was legal. (It gets treated as more arguments to the `ECHO` command.) But the second batch file puts the extra junk in a place where junk is not legal, and so it is rejected as a syntax error.

If you take a step back, you'll see that the real problem is that the batch file uses a path with spaces but fails to quote it properly.

Whose idea was it to allow redirection operators to appear anywhere in the line? Answer: Ken Thompson, in version 2 of sh. If you think this is a stupid feature, you can let him know.

Raymond Chen

Follow

