The redirection can come anywhere on the line, and you can use that to get rid of the spaces

devblogs.microsoft.com/oldnewthing/20170801-00

August 1, 2017

Raymond Chen

We saw last time that the redirection can come anywhere on the line, and we saw that unquoted paths with embedded spaces can seem to work, but in fact doesn't. And you don't notice because the command processor is not parsing the command the way you think.

Another problem with redirection is the unwanted trailing spaces. If you say

```
echo foo >result.txt
```

then the file `result.txt` consists of six characters:

`f`	`o`	`o`		`\r`	`\n`

Notice that there's a space before the CR+LF. You probably didn't want that space, but it's there because you put a space before the redirection operator. The parser takes out the redirection operator and the file name, but the space before the redirection operator is still there.

You can remove the space by squishing the redirection operator right against the string you want to print.

```
echo foo>result.txt
```

Now the resulting file is

`f`	`o`	`o`	`\r`	`\n`

Cool. No stray space.

And then later you get a bug because this string didn't get saved to the file properly:

```
set MESSAGE=2 for 1
echo %MESSAGE%>result.txt
```

The resulting file is

`2`		`f`	`o`	`r`		`\r`	`\n`

Where did the `1` go?

Another part of the syntax for redirection is that if you put a number in front of the redirection operator, it specifies which file descriptor you want to redirect. In practice, this number is nearly always 2 (stderr), because 1 (stdout) is the default, and 0 is stdin, which nobody writes to. But it means that the expansion of the command line becomes

```
echo 2 for 1>result.txt
```

and you are now performing an explicit redirection of stdout. The `1` is redundant because the default is stdout, but hey, you asked for it.

To get rid of this space, you can take advantage of the principle in the subject of today's post. Put the redirection operator somewhere else. I like to put it at the front.

```
>result.txt echo %MESSAGE%
```

Putting the redirection at the front also makes building a multi-line document easier to read.

```
rem old way
echo first line>result.txt
echo second line>>result.txt
echo third line>>result.txt

rem new way
 >result.txt echo first line
>>result.txt echo second line
>>result.txt echo third line
```

Raymond Chen

Follow