If you're not using the command line interpreter, then the command line interpreter metacharacters mean nothing

devblogs.microsoft.com/oldnewthing/20130718-00

July 18, 2013

Raymond Chen

A customer observed that the parameters passed to `CreateProcess` were not being interpreted correctly.

```
char commandLine[] = "reg.exe query "
                     "HKLM\\SYSTEM\\CurrentControlSet\\"
                     "Enum\\HID\VID_0461^&PID_4D15";
CreateProcess(NULL, commandLine, ...);
```

The process is created successfully, but it prints the message `ERROR: The system was unable to find the specified registry key or value.`. Why aren't the parameters being parsed correctly by `CreateProcess`? They work fine if I paste them into a command prompt.

This is a variation of the problem we saw a few years ago. Back then, we had a string with command line redirection metacharacters, but since we were passing them directly to `CreateProcess`, the command interpreter never got a chance to interpret them. Here, we have a string that contains an ampersand, which has special meaning to the command interpreter, so we escaped it so that the command interpreter won't try to treat it as command separator. But then we passed it directly to the `CreateProcess` function without ever invoking the command processor.

It's like exchanging your U.S. dollars for Canadian dollars because you think you're going to drive through Canada, and then deciding to take the southern route instead, and then wondering why the Canadian money doesn't work. You went to the extra effort of converting the string into a form that will survive a journey you never sent it on!

If you're going to munge the string so that you get the desired end result after it travels through the command interpreter, then you need to send it through the command interpreter. Or more preferably, cut out the middle man and don't bother munging it in the first place.

If you're going to prepare a string for a journey, you need to send it on that journey.

Raymond Chen

Follow