How can I count how many calls of a cmd file?

Asked 15 years, 9 months ago Modified 14 years, 11 months ago

Viewed 951 times

1

How can I count how many calls of a cmd file?

I'm struggling with something like this but it didn't work:

```
@IF NOT EXIST Calls.log echo. > Calls.log
@for  %%i in (Calls.log) do set size=%%~zi
@IF %size% EQU 0 (
@ECHO 1 > Calls.log
) ELSE (
@set /p v=<Calls.log
@set /A v+=1
@echo %v% > Calls.log
)
```

scripting batch-file cmd

Share

Improve this question

Follow

edited Jan 25, 2010 at 13:26

Peter Mortensen

31.6k ● 22 ● 109 ● 133

asked Mar 16, 2009 at 23:32

Hassan

3 Answers

Sorted by: Highest score (default)

2

If all you're trying to do is count how many times a cmd script is called, you can just append one character to a file every time it runs rather than fiddling around with expression evaluation every time the script is run. This also has the advantage of making the script quicker since the analysis of the count is moved elsewhere.

The counter file expands by one byte every time, so watch out if you're calling it a truly large number of times since the file will expand. But even, calling it once per second, a 1G file will accrue only after 30 years.

At the top of your script, just put:

```
set >>countfile.txt <nul: /p x=X
```

This simply adds the character `X` to the end of the `countfile.txt` file every time the script is called. It uses the "`set/p`" command with input/output redirection, which is the only way I'm aware of to get a character out without a CR/LF following it (like the UNIX "`echo -n`").

To get a count of the number of calls to date, you can use the file size environment variable modifier, as in the following script (I expect this will be done less often than running the script so it's better to put the grunt work here [in fact, it's not a lot of grunt work since it's not counting the characters, rather it gets the information directly from the directory entry]):

```
    @echo off
    goto :main
:getsize
    echo %~z1
    goto :eof
:main
    setlocal enableextensions
enabledelayedexpansion
    call :getsize countfile.txt
    endlocal
```

To reset the count, use the following extremely complicated command (I'm thinking of patenting this):

```
del countfile.txt
```

One other thing I'd suggest - you don't need to prefix every command with "`@`" to prevent echo, you can simply put "`@echo off`" at the top of your script for global no-echo. If you want to selectively echo some commands, just ignore this paragraph.

Share Improve this answer Follow

answered Mar 17, 2009 at 1:12
paxdiablo
880k ● 241 ● 1.6k ● 2k

1

The complete `if` block is parsed at once and thus all environment variables in it are getting replaced by their values at the time *before* the `if` block gets executes. You need to enable delayed variable expansion and use `!v!`:

```
@setlocal enabledelayedexpansion
@IF NOT EXIST Calls.log echo. > Calls.log
@for  %%i in (Calls.log) do set size=%%~zi
@IF %size% EQU 0 (
    @ECHO 1 > Calls.log
) ELSE (
    @set /p v=<Calls.log
    @set /A v+=1
    @echo !v! > Calls.log
)
```

And you can simplify the code as follows:

```
@echo off
setlocal enabledelayedexpansion
IF NOT EXIST Calls.log (
    ECHO 1 > Calls.log
) ELSE (
    set /p v=<Calls.log
    set /A v+=1
    echo !v! > Calls.log
)
```

There is no need to create the file beforehand (and even that I'd solve with `copy nul Calls.log`, since that ensures a file size of 0).

Share Improve this answer Follow

answered Mar 17, 2009 at 0:32

Joey
354k • 86 • 698 • 694

0

The following code works on my computer:

```
@if not exist Calls.log (
echo 0 > Calls.log
)
@set /p v=< Calls.log
@set /a v=v+1
echo %v% > Calls.log
```

Share Improve this answer Follow

answered Mar 17, 2009 at 0:48

user64349

603 ● 1 ● 5 ● 8