Any way to write a Windows .bat file to kill processes? [closed]

Asked 16 years, 3 months ago   Modified 4 years, 6 months ago

Viewed 453k times

148

Every time I turn on my company-owned development machine, I have to kill 10+ processes using the Task Manager or any other process management app just to get decent performance out of my IDE. Yes, these are processes from programs that my company installs on my machine for security and compliance. What I'd like to do is have a .bat file or script of some kind with which I can kill the processes in question.

Does anybody know how to do this?

Share

Improve this question

Follow

edited Jul 10, 2019 at 16:55

bubbleking
3,551 • 4 • 31 • 56

asked Aug 29, 2008 at 2:09

codeLes
3,069 • 3 • 29 • 27

---

1   This should be handy also when in an automation environment and a run away subprocess needs to be terminated if it fails to finish or respond in a decent amount of time. – Danny Staple Jun 17, 2015 at 15:46

---

1   I posted this question link to all my direct colleages because we are roundabout 20 S/W developers in a company with roundabout 2000 engineers (doing CAD). We are permanently struggling with the IT administration department because (due to security policy) we are nearly unable to do our work... (It doesn't help technically but it's nice to hear that other S/W developers seem to have similar problems.) – Scheff's Cat May 14, 2017 at 8:11

---

9 Answers

Sorted by: Highest score (default) ⇕

231

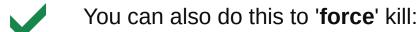You can do this with '**taskkill**'. With the /IM parameter, you can specify image names.

Example:

```
taskkill /im somecorporateprocess.exe
```

You can also do this to '**force**' kill:

Example:

```
taskkill /f /im somecorporateprocess.exe
```

Just add one line per process you want to kill, save it as a .bat file, and add in your startup directory. Problem solved!

If this is a legacy system, PsKill will do the same.

Share   Improve this answer

Follow

edited May 30, 2020 at 1:28

Nith
3 ● 2

answered Aug 29, 2008 at 2:14

Factor Mystic
26.7k ● 16 ● 81 ● 96

---

13   I had success using `taskkill /F /IM <processname.exe> /T` to force closing of *all* processes by their name. – Benj Aug 19, 2015 at 9:15 ✎

114

```
taskkill /f /im "devenv.exe"
```
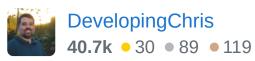
this will forcibly kill the pid with the exe name "devenv.exe"

equivalent to -9 on the nix'y kill command

Share Improve this answer

Follow

answered Aug 29, 2008 at 2:13

DevelopingChris
40.7k ● 30 ● 89 ● 119

11

As TASKKILL might be unavailable on some Home/basic editions of windows here some alternatives:

```
TSKILL processName
```

or

```
TSKILL PID
```

Have on mind that `processName` should not have the `.exe` suffix and is limited to 18 characters.

Another option is `WMIC` :

```
wmic Path win32_process Where "Caption Like
'MyProcess%.exe'" Call Terminate
```

wmic offer even more flexibility than taskkill with its SQL-like matchers .With `wmic Path win32_process get` you can see the available fileds you can filter (and `%` can be used as a wildcard).

Share Improve this answer

Follow

edited Jan 24, 2020 at 9:31

answered Feb 2, 2015 at 8:22

npocmaka
57.2k ● 18 ● 161 ● 193

I'm assuming as a developer, you have some degree of administrative control over your machine. If so, from the command line, run msconfig.exe. You can remove many

5

processes from even starting, thereby eliminating the need to kill them with the above mentioned solutions.

Share   Improve this answer

Follow

answered Aug 29, 2008 at 3:18

Jason Z
13.4k ● 15 ● 51 ● 62

Get [Autoruns](#) from Mark Russinovich, the Sysinternals guy that discovered the Sony Rootkit... Best software I've ever used for cleaning up things that get started automatically.
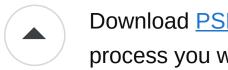
5

Share   Improve this answer

Follow

answered Aug 29, 2008 at 3:45

Brian Stewart
9,267 ● 11 ● 55 ● 66

Download [PSKill](#). Write a batch file that calls it for each process you want dead, passing in the name of the process for each.

4

Share   Improve this answer

Follow

answered Aug 29, 2008 at 2:12

Shog9
159k ● 36 ● 235 ● 240

Use Powershell! Built in cmdlets for managing processes. [Examples here](#) (hard way), [here](#)(built in) and [here](#) (more).

1

Share Improve this answer

Follow

Please find the below logic where it works on the condition.

If we simply call `taskkill /im applicationname.exe`, it will kill only if this process is running. If this process is not running, it will throw an error.

So as to check before `takskill` is called, a check can be done to make sure execute `taskkill` will be executed only if the process is running, so that it won't throw error.

```
tasklist /fi "imagename eq applicationname.exe"
|find ":" > nul

if errorlevel 1 taskkill /f /im
"applicationname.exe"
```

Share Improve this answer

Follow

1 easier to just redirect the error message of `taskkill` to `nul` – Stephan Oct 26, 2016 at 12:39

I add a very small edit in the end and it works like a charm: tasklist /fi "imagename eq applicationname.exe" |find ":" > nul

if errorlevel 1 taskkill /f /im "applicationname.exe" > null

– techsystem2000 Oct 24, 2021 at 4:37 ✎

Here I wrote an example command that you can paste in your `cmd` command line prompt and is written for `chrome.exe`.

```
FOR /F "tokens=2 delims= " %P IN ('tasklist /FO
Table /M "chrome*" /NH') DO (TASKKILL /PID %P)
```

The for just takes all the `PID`s listed on the below `tasklist` command and executes `TASKKILL /PID` on every `PID`

```
tasklist /FO Table /M "chrome*" /NH
```

If you use the for in a batch file just use %%P instead of %P

Share Improve this answer

Follow

answered Feb 12, 2018 at 21:58

Eduard Florinescu
17.5k ● 29 ● 121 ● 186