

WINDOWS SCM – KILLING A WINDOWS SERVICE THAT SEEMS TO HANG ON “STOPPING”

It sometimes happens (and it's not a good sign most of the time): you'd like to stop a Windows Service, and when you issue the stop command through the **SCM (Service Control Manager)** or by using the **ServiceProcess** classes in the .NET Framework or by other means (net stop, **Win32 API**), the service remains in the state of "stopping" and never reaches the stopped phase.

It's pretty simple to simulate this behavior by creating a **Windows Service** in **C#** (or any **.NET** language whatsoever) and adding an infinite loop in the Stop method. The only way to stop the service is by killing the process then.

However, sometimes it's not clear what the process name or **ID** is (e.g. when you're running a service hosting application that can cope with multiple instances such as **SQL Server Notification Services**). The way to do it is as follows:

- Go to the *command-prompt* and query the service (e.g. the **SMTP** service) by using sc:

```
sc queryex SMTPSvc
```

- This will give you the following information:

```
SERVICE_NAME: SMTPSvc
TYPE : 20 WIN32_SHARE_PROCESS
STATE : 4 RUNNING
(STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x0
PID : 388
FLAGS :
```

- Or something like this (the "state" will mention stopping).
- Over here you can find the process identifier (**PID**), so it's pretty easy to kill the associated process either by using the task manager or by using *taskkill*:

```
taskkill /PID 388 /F
```

Please be careful when you do this; it's useful for emergencies but you shouldn't use it on a regular basis (*use it as a last chance to solve the problem or to avoid the need of a reboot in an exceptional situation*).

It can even be used to stop a service that has the "**NOT-STOPPABLE**" and/or "**IGNORES_SHUTDOWN**" flag set (e.g. **Terminal Services** on a **Windows Server 2003** is non-stoppable), at least when it's not hosted in the system process.

You can query all this information by means of the sc command.

Where the **/F** flag is needed to force the process kill (*first try without the flag*).