How to Stop a Service with PowerShell

lazyadmin.nl/powershell/stop-service

October 25, 2024

Services run continuously in the background on computers. But sometimes you need to stop a service before you can perform a specific task. We can do this with the Stop-Service cmdlet in PowerShell.

The cmdlet comes with a few options like forcing the stop of a service, stopping multiple services based on a part of the name, and the no wait option that allows you script to continue while the service is being stopped.

In this article, we will look at the different options for stopping a service with PowerShell.

Stop Service with PowerShell

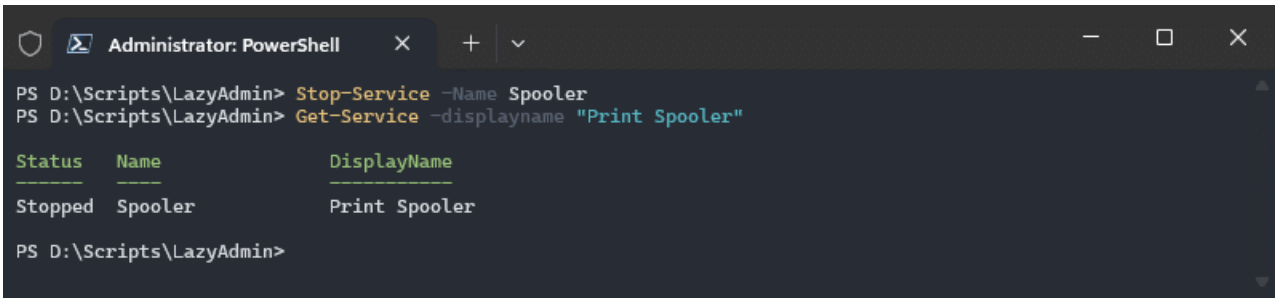
To stop a service we need to know the name of the service. The name is often **not the same** as the display name from a service. When you would open the services.msc console, then the name listed there is the display name, not the service name.

For example, when we look at the “Print Spooler” service, the service name of it is actually “Spooler”. We can also see this when we look up the service with the `Get-Service` cmdlet:

```
Get-Service -displayname "Print Spooler"
# Result
Status Name DisplayName
------ ---- -----------
Running Spooler Print Spooler
```

So to stop the service in this case, you have three options. You can stop it using the actual name of the service, use the `-displayname` parameter or pipe the `Stop-Service` cmdlet behind the `Get-Service` cmdlet:

```
# Stop service by it's name
Stop-Service "Spooler"
# Stop service based on the display name
Stop-Service -Display "Print Spooler"
# Get service first, then stop it
Get-Service -displayname "Print Spooler" | Stop-Service
```



Using Wildcards

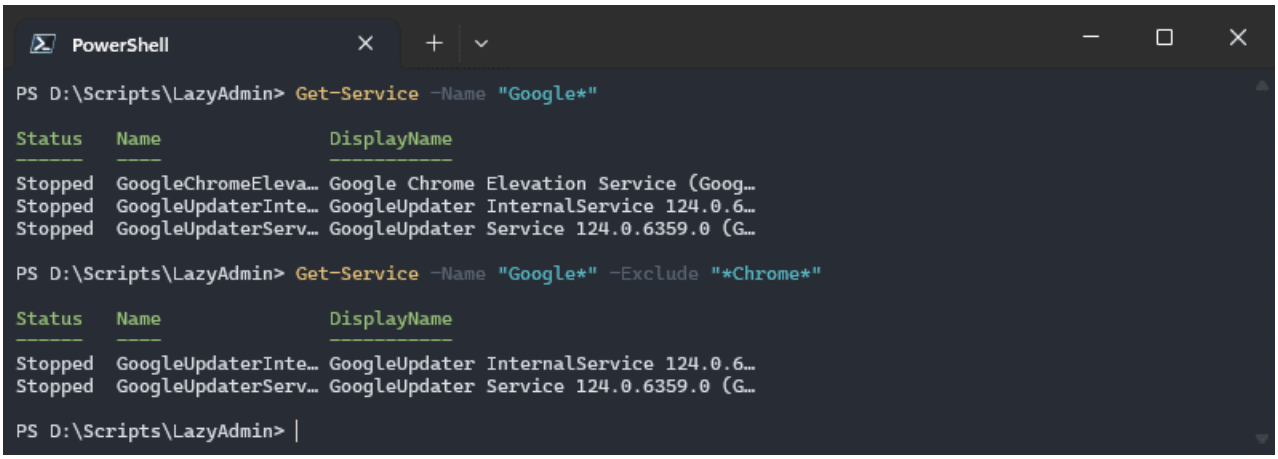
If you don't know the exact name of the service, then you can also use a wildcard to find the service that you want to stop.

This method also allows you to stop multiple services that contain the same name with a single command. For example, to stop all Google services, we first get all services where the name starts with Google, and then stop all of them using the stop-service cmdlet:

```
# Stop all services where the name starts with Google
Stop-Service -Name "Google*" | Stop-Service
```

When using wildcards, you sometimes want or need to exclude specific services. We can do this with the `-exclude` parameter, which also supports wildcards. This way we can for example get all Google services, except the Chrome services:

```
Stop-Service -Name "Google*" -Exclude "*Chrome*"
```



You can also use the `-Include` parameter, so for the example above, you could get all get all services where the name contains Google and include "Update" to get only the updater services.

Force Stop a Service

Some services don't like to be stopped. The service might be working on a task or has dependent services, preventing you from stopping it.

To solve this, we can use the `-Force` parameter. This will stop a service even if it has dependent services.

Stop-Service -Name "iisadmin" -Force
When using the force option to stop services that have dependent services, it's a good idea to use the `-Confirm` option as well. This way you can check and confirm for each dependent service if you want to stop it as well.

NoWait

When you stop a service with PowerShell, your script will wait until the service is actually stopped before your script continues. In most cases, this is fine and the recommended way to do it, but sometimes you have a service that just takes quite long to stop and your script doesn't need to wait for it.

For these cases, you can use the `-NoWait` option. This will tell PowerShell that it can continue with the rest of the script while the service is being stopped:

Stop-Service -Name "Spooler" -NoWait

Stop Service on Remote Computer

To stop a service on a remote computer with PowerShell, we first need to get the service on the remote computer with the Get-Service cmdlet and then we can pipe the stop-service cmdlet behind it.

The `Get-Service` cmdlet comes with built-in support for remote computers with the `-Computername` parameter. Important to note here, that this method **doesn't work on PowerShell 7** and higher

Get-Service -ComputerName $hostname -Name "Spooler" | Stop-Service
When using PowerShell 7, we will need to use the Invoke-Command cmdlet instead. This cmdlet allows you to run scripts or commands on remote computers. To stop a service on a remote computer with PowerShell 7, we can do the following:

Invoke-Command -ComputerName $hostname -ScriptBlock { Stop-Service -Name "Spooler" }

Wrapping Up

Stopping a service with PowerShell is not that hard, just make sure that you use the correct service name or use the display name parameter.

I hope this article helped you, if you have any questions, just drop a comment below.

I hate spam to, so you can unsubscribe at any time.