

Using the PowerShell Restart-Service CmdLet on Remote Servers

✘ I pride myself on being an lazy admin. The preferred term is efficient, but what I mean by “lazy” is that I like to do the least amount of typing to achieve the desired result. Another key about my technique is that while I can use Remote Desktop or the Computer Management MMC to manage servers, I prefer to do as much through the command line as possible.

PowerShell remoting has just what I need to handle my task today. I have to restart a Windows Service on my remote server with limited effort. In the past I would use the **sc.exe** Resource Kit utility to query and manage services on remote servers:

```
SC \DC1 QUERY
```



The results can be pared down using find or the Win32 build of grep which I use to go after specific services. Once I know the service name I can run two commands to stop and start the service:

```
SC \DC1 STOP MyServiceName
```

```
SC \DC1 START MyServiceName
```

Since I’m moving my day-to-day processes to PowerShell and PowerCLI wherever possible, now I can leverage the **Get-Service**, **Stop-Service**, **Start-Service** and most importantly the **Restart-Service** CmdLets.

There are two main reasons for the switch. Firstly, it’s a direction shift to use the native supported language for all my Windows based devices. Secondly, if you use a PowerShell console window you see that using SC.EXE is no longer possible because SC is an alias for the Set-Content CmdLet.



Where I used to have to run the stop and start commands individually, now I can use the **Restart-Service** PowerShell CmdLet which will stop, wait and start the service for me without having to be monitored. Even more exciting is that if I have a few services that are running, I can use the pipeline to run the **Restart-Service** against more than one service in a simple one-liner.

We are faced immediately with a limitation of the **Restart-Service**. While this CmdLet works great when you are logged in locally, there is no option to run it against a remote server. It is implied and required that we are running it in a local session on the device. But our goal was to eliminate the need to be remotely attached to the server via RDP so we need to add one more quick step.

The solution is to use PowerShell remote sessions with the **Enter-PSSession** CmdLet. First we will launch our local PowerShell command prompt (perhaps you want to use my multi-environment PowerShell console). From within that console we simply initiate a remote PowerShell session using this:

Enter-PSSession DC1



Now you will see that we are able to run PowerShell CmdLets as if we were on the DC1 console directly. This allows us to use the Restart-Service CmdLet now against the desired services on DC1 directly. I want to restart any services which begin with **dfs** so I can use the wildcard asterisk (*) when sending the service name to the CmdLet.

Note that your command prompt shows the active server on the left which in this case it reads [dc1]:



Make sure that for cleanliness that you also issue the Exit-PSSession command to exit and free up the remote PowerShell session to the server:



I hope that you can save some time and effort with technique. I know that I certainly do.

Here are the TechNet articles for the CmdLets we worked with:

Stop-Service : <http://technet.microsoft.com/en-us/library/ee177005.aspx>

Start-Service : <http://technet.microsoft.com/en-us/library/ee177001.aspx>

Restart-Service : <http://technet.microsoft.com/en-us/library/ee176942.aspx>

Enter-PSSession : <http://technet.microsoft.com/en-us/library/hh849707.aspx>

Exit-PSSession : <http://technet.microsoft.com/en-us/library/hh849743.aspx>