Powershell Lab Chapter 4

Alex Stock

**1.  What happens if you run *Get-Service | Export-CSV services.csv | Out-File* from the console? Why does that happen?**

PS> Get-Service | Export-CSV services.csv | Out-File

Out-File : Cannot process argument because the value of argument "path" is null. Change the value of argument "path" to a non-null value.

The commandlet “Out-File” has a compulsory parameter named “FilePath”.

PS> Get-Help Out-File -Parameter FilePath

-FilePath <String>

Specifies the path to the output file.

Required? true

Position? 1

Default value

Accept pipeline input? false

Accept wildcard characters? False

This parameter is required, but is not provided in the above example. Further, the “Out-File” commandlet is not required. “Out-File” saves the command output to a file, but the “Export-CSV” commandlet already performs this function.

**2.  Apart from getting one or more services and piping them to *Stop-Service*, what other means does *Stop-Service* provide for you to specify the service or services you want to stop? Is it possible to stop a service without using *Get-Service* at all?**

The “Name” parameter allows you to specify the Service Name of the services you want to stop. These Service Names can be passed through the command pipe. The “DisplayName” parameter allows you to specify the Display Names of the services to be stopped. Both of these parameters can be used to stop services without using the “Get-Service” commandlet.

PS> Stop-Service -Name DNS -WhatIf

What if: Performing the operation "Stop-Service" on target "DNS Server (DNS)".

PS> Stop-Service -DisplayName "DNS Server" -WhatIf

What if: Performing the operation "Stop-Service" on target "DNS Server (DNS)".

**3.  What if you wanted to create a pipe-delimited file instead of a comma-separated file? You would still use the *Export-CSV* command, but what parameters would you specify?**

The “Delimiter” parameter allows you to create a pipe-delimited file.

Get-Service | Export-CSV Service.csv -Delimiter "|"

**4.  Is there a way to eliminate the *#* comment line from the top of an exported CSV file? That line normally contains type information, but what if you wanted to omit that from a particular file?**

The “NoTypeInformation” parameter will cause the export-CSV commandlet to omit the type information from the CSV file.

**5.  *Export-CliXML* and *Export-CSV* both modify the system, because they can create and overwrite files. What parameter would prevent them from overwriting an existing file? What parameter would ask you if you were sure before proceeding to write the output file?**

The “NoClobber” parameter will prevent the export-CSV and export-CliXML commandlets from overwriting an existing file.

**6.  Windows maintains several regional settings, which include a default list separator. On U.S. systems, that separator is a comma. How can you tell *Export-CSV* to use the system’s default separator, rather than a comma?**

The “UseCulture” parameter will cause the export-CSV commandlet to use the system’s default separator.