



python



PowerShell

FPDF Library
PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Export-Counter'

PS C:\Users\wahid> Get-Help Export-Counter

NAME

Export-Counter

SYNOPSIS

Exports performance counter data to log files.

SYNTAX

```
Export-Counter [-Path] <System.String> [-Circular] [-FileFormat  
<System.String>] [-Force] -InputObject  
<Microsoft.PowerShell.Commands.GetCounter.PerformanceCounterSampleSet[]>  
[-MaxSize <System.UInt32>] [<CommonParameters>]
```

DESCRIPTION

The `Export-Counter` cmdlet exports performance counter data (PerformanceCounterSampleSet objects) to log files in binary performance log (`.blg`), comma-separated value (`.csv`), or tab-separated value (`.tsv`) format. You use this cmdlet to log performance counter data.

The `Export-Counter` cmdlet is designed to export data that is returned by the

`Get-Counter` and `Import-Counter` cmdlets.

This cmdlet runs only on Windows 7, Windows Server 2008 R2, and later versions of Windows.

PARAMETERS

`-Circular <System.Management.Automation.SwitchParameter>`

Indicates that the output file is a circular log with first in, first out (FIFO) format. When you include this parameter, the `MaxSize` parameter is required.

`-FileFormat <System.String>`

Specifies the output format of the output log file.

The acceptable values for this parameter are:

- `CSV`

- `TSV`

- `BLG`

The default value is `BLG`.

`-Force <System.Management.Automation.SwitchParameter>`

Overwrites and replaces an existing file if one exists in the location specified by the `Path` parameter.

`-InputObject`

`<Microsoft.PowerShell.Commands.GetCounter.PerformanceCounterSampleSet[]>`

Specifies, as an array, the counter data to export. Enter a variable that contains the data or a command that gets the data, such as the ``Get-Counter`` or ``Import-Counter`` cmdlet.

`-MaxSize <System.UInt32>`

Specifies the maximum size of the output file in megabytes (MB).

If the `Circular` parameter is specified, then when the log file reaches the specified maximum size, the oldest entries are deleted as newer ones are added. If the `Circular` parameter is not specified, then when the log file reaches the specified maximum size, no new data is added and the cmdlet generates a non-terminating error.

`-Path <System.String>`

Specifies the path and file name of the output file. Enter a relative or absolute path on the local computer, or a Uniform Naming Convention (UNC) path to a remote computer, such as ``\Computer\Share\file.blg``. This parameter is required.

The file format is determined by the value of the `FileFormat` parameter, not by the file name extension in the path.

`<CommonParameters>`

This cmdlet supports the common parameters: `Verbose`, `Debug`, `ErrorAction`, `ErrorVariable`, `WarningAction`, `WarningVariable`, `OutBuffer`, `PipelineVariable`, and `OutVariable`. For more information, see `about_CommonParameters` (<https://go.microsoft.com/fwlink/?LinkID=113216>).

----- EXAMPLE 1: Export counter data to a file -----

```
Get-Counter "\Processor(*)\% Processor Time" | Export-Counter -Path  
$HOME\Counters.blg
```

The command uses the `Get-Counter` cmdlet to collect processor time data. It uses a pipeline operator (`|`) to send the data to the `Export-Counter` cmdlet. The `Export-Counter` command uses the Path variable to specify the output file.

Because the data set might be very large, this example sends the data to `Export-Counter` through the pipeline. If the data were saved in a variable, you might use a disproportionate amount of memory.

----- Example 2: Export a file to a counter file format -----

```
$1GBInMB = 1024 # 1GB = 1024MB
```

```
Import-Counter Threads.csv | Export-Counter -Path ThreadTest.blg -Circular  
-MaxSize $1GBInMB
```

Example 3: Get counter data from a remote computer and save the data to a file

```
$C = Get-Counter -ComputerName Server01 -Counter "\Process(*)\Working Set -  
Private" -MaxSamples $C | Export-Counter -Path \\Server01\Perf\WorkingSet.blg
```

20

----- Example 4: Re-log existing data -----

```
$All = Import-Counter DiskSpace.blg
```

```
$LowSpace = $All | Where-Object {$_.CounterSamples.CookedValue -lt 15}
```

```
$LowSpace | Export-Counter -Path LowDiskSpace.blg
```

REMARKS

To see the examples, type: "get-help Export-Counter -examples".

For more information, type: "get-help Export-Counter -detailed".

For technical information, type: "get-help Export-Counter -full".

For online help, type: "get-help Export-Counter -online"