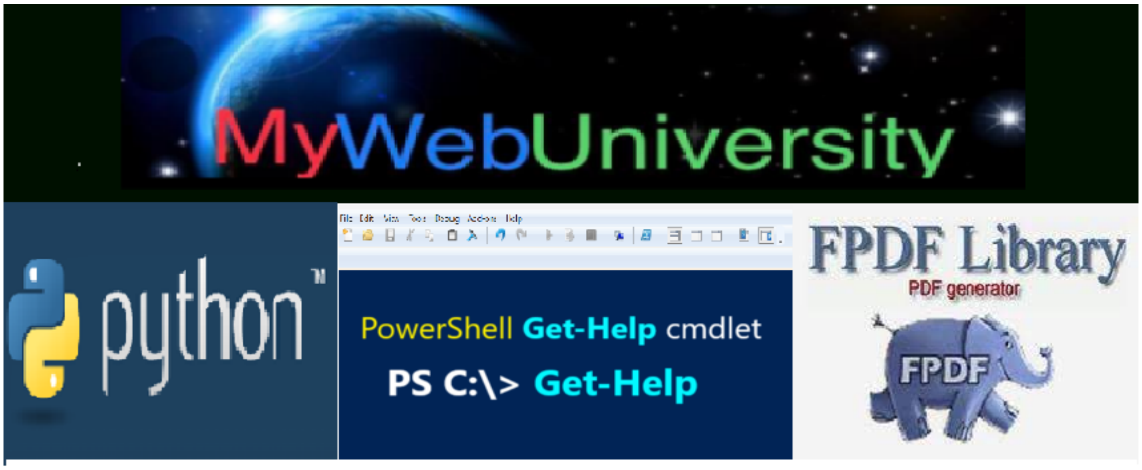

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Measure-Object'

PS:\>Get-HELP Measure-Object -Full

NAME

Measure-Object

SYNOPSIS

Calculates the numeric properties of objects, and the characters, words, and lines in string objects, such as files of text.

SYNTAX

Measure-Object [[-Property] <System.String[]>] [-Average] [-InputObject <System.Management.Automation.PSObject>] [-Maximum] [-Minimum] [-Sum] [<CommonParameters>]

Measure-Object [[-Property] <System.String[]>] [-Character] [-IgnoreWhiteSpace] [-InputObject <System.Management.Automation.PSObject>] [-Line] [-Word]

[<CommonParameters>]

DESCRIPTION

The `Measure-Object` cmdlet calculates the property values of certain types of object. `Measure-Object` performs three types of measurements, depending on the

parameters in the command.

The `Measure-Object` cmdlet performs calculations on the property values of objects. You can use `Measure-Object` to count objects or count objects with a specified

Property . You can also use `Measure-Object` to calculate the Minimum , Maximum , Sum , StandardDeviation and Average of numeric values. For String objects, you can

also use `Measure-Object` to count the number of lines, words, and characters.

PARAMETERS

-Average <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet displays the average value of the specified properties.

```
Required?                    false
Position?                    named
Default value                False
Accept pipeline input?       False
Accept wildcard characters?  false
```

-Character <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet counts the number of characters in the input objects.

> [!NOTE] > The Word , Char and Line switches count inside each input object, as well as across > input objects. See Example 7.

```
Required?                    false
Position?                    named
Default value                False
Accept pipeline input?       False
Accept wildcard characters?  false
```

-IgnoreWhiteSpace <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet ignores white space in character counts. By default, white space is not ignored.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-InputObject <System.Management.Automation.PSObject>

Specifies the objects to be measured. Enter a variable that contains the objects, or type a command or expression that gets the objects.

When you use the InputObject parameter with `Measure-Object`, instead of piping command results to `Measure-Object`, the InputObject value is treated as a single

object.

It is recommended that you use `Measure-Object` in the pipeline if you want to measure a collection of objects based on whether the objects have specific values

in defined properties.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   True (ByValue)

Accept wildcard characters? false

-Line <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet counts the number of lines in the input objects.

> [!NOTE] > The Word , Char and Line switches count inside each input object, as well as across > input objects. See Example 7.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Maximum <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet displays the maximum value of the specified properties.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Minimum <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet displays the minimum value of the specified properties.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Property <System.String[]>

Specifies one or more properties to measure. If you do not specify any other measures, `Measure-Object` counts the objects that have the properties you specify.

Required?   false

Position?   0

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-Sum <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet displays the sum of the values of the specified properties.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Word <System.Management.Automation.SwitchParameter>

Indicates that the cmdlet counts the number of words in the input objects.

> [!NOTE] > The Word , Char and Line switches count inside each input object, as well as across > input objects. See Example 7.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

<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).

INPUTS

System.Management.Automation.PSObject

You can pipe objects to this cmdlet.

OUTPUTS

Microsoft.PowerShell.Commands.GenericMeasureInfo

By default, this cmdlet returns a GenericMeasureInfo object.

Microsoft.PowerShell.Commands.TextMeasureInfo

When you use the Word parameter, this cmdlet returns a TextMeasureInfo object.

NOTES

Windows PowerShell includes the following aliases for `Measure-Object`:

- `measure`

---- Example 1: Count the files and folders in a directory ----

```
Get-ChildItem | Measure-Object
```

--------- Example 2: Measure the files in a directory ---------

```
Get-ChildItem | Measure-Object -Property length -Minimum -Maximum -Sum -Average
```

------------ Example 3: Measure text in a text file ------------

```
"One", "Two", "Three", "Four" | Set-Content -Path C:\Temp\tmp.txt
Get-Content C:\Temp\tmp.txt | Measure-Object -Character -Line -Word

Lines Words Characters Property
----- ----- ---------- --------
    4     4         15
```

-- Example 4: Measure objects containing a specified Property --

```
$services = Get-Service

$processes = Get-Process

$services + $processes | Measure-Object

$services + $processes | Measure-Object -Property DisplayName


Count    : 682

Average  :

Sum      :

Maximum  :

Minimum  :

Property :


Count    : 290

Average  :

Sum      :

Maximum  :

Minimum  :

Property : DisplayName
```

-------- Example 5: Measure the contents of a CSV file --------

```
Import-Csv d:\test\serviceyrs.csv | Measure-Object -Property years -Minimum -Maximum -Average
```

-------------- Example 6: Measure Boolean values --------------

```
Get-ChildItem | Measure-Object -Property psiscontainer -Maximum -Sum -Minimum -Average


Count          : 126

Average          : 0.0634920634920635

Sum            : 8
```

```
Maximum           : 1

Minimum           : 0

StandardDeviation :

Property          : PSIsContainer
```

------------------ Example 7: Measure strings ------------------

```
# The newline character `n separates the string into separate lines, as shown in the output.

"One`nTwo`nThree"

"One`nTwo`nThree" | Measure-Object -Line


One

Two

Three



Lines Words Characters Property

----- ----- ---------- --------

    3


# The first string counts as a single line.

# The second string is separated into two lines by the newline character.

"One", "Two`nThree" | Measure-Object -Line


Lines Words Characters Property

----- ----- ---------- --------

    3


# The Word switch counts the number of words in each InputObject

# Each InputObject is treated as a single line.

"One, Two", "Three", "Four Five" | Measure-Object -Word -Line
```

```
Lines Words Characters Property
----- ----- ---------- --------
    3     5
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/measure-object?view=powershell-5.1&WT.mc_id=ps-gethelp

Compare-Object

ForEach-Object

Group-Object

New-Object

Select-Object

Sort-Object

Tee-Object

Where-Object