PowerShell Get-Help on command 'Format-List'

PS C:\Users\wahid> Get-Help Format-List

NAME

    Format-List

SYNOPSIS

    Formats the output as a list of properties in which each property appears on a
    new line.

SYNTAX

    Format-List [[-Property] <System.Object[]>] [-DisplayError] [-Expand {CoreOnly
    | EnumOnly | Both}] [-Force] [-GroupBy <System.Object>] [-InputObject
    <System.Management.Automation.PSObject>] [-ShowError] [-View <System.String>]
    [<CommonParameters>]

DESCRIPTION

    The `Format-List` cmdlet formats the output of a command as a list of
    properties in which each property is displayed on a separate line. You can use
    `Format-List` to format and display all or selected properties of an object as
    a list (`Format-List -Property *`).

Because more space is available for each item in a list than in a table, PowerShell displays more properties of the object in the list, and the property values are less likely to be truncated.

PARAMETERS

-DisplayError <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet displays errors at the command line. This parameter is rarely used, but can be used as a debugging aid when you are formatting expressions in a `Format-List` command, and the expressions do not appear to be working.

-Expand <System.String>

Specifies the formatted collection object, as well as the objects in the collection. This parameter is designed to format objects that support the System.Collections.ICollection interface. The default value is `EnumOnly`. The acceptable values for this parameter are:

- `EnumOnly`. Displays the properties of the objects in the collection.

- `CoreOnly`. Displays the properties of the collection object.

- `Both`. Displays the properties of the collection object and the properties of objects in the

collection.

-Force <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet displays all the error information. Use with the DisplayError or ShowError parameter. By default, when an error object is written to the error or display streams, only some error information is displayed.

Also required when formatting certain .NET types. For more information, see the Notes (#notes)section.

-GroupBy <System.Object>

Specifies the output in groups based on a shared property or value. Enter an expression or a property of the output.

The value of the GroupBy parameter can be a new calculated property. The calculated property can be a script block or a hash table. Valid key-value pairs are:

- `Name` (or `Label`) - `<string>`

- `Expression` - `<string>` or `<script block>`

- `FormatString` - `<string>`

For more information, see about_Calculated_Properties (../Microsoft.PowerShell.Core/About/about_Calculated_Properties.md).

-InputObject <System.Management.Automation.PSObject>

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

-Property <System.Object[]>

Specifies the object properties that appear in the display and the order in which they appear. Wildcards are permitted.

If you omit this parameter, the properties that appear in the display depend on the object being displayed. The parameter name Property is optional. You cannot use the Property and View parameters in the same command.

The value of the Property parameter can be a new calculated property. The calculated property can be a script block or a hash table. Valid key-value pairs are:

- `Name` (or `Label`) - `<string>`

- `Expression` - `<string>` or `<script block>`

- `FormatString` - `<string>`

For more information, see about_Calculated_Properties (../Microsoft.PowerShell.Core/About/about_Calculated_Properties.md).

-ShowError <System.Management.Automation.SwitchParameter>
Indicates that the cmdlet sends errors through the pipeline. This parameter is rarely used, but can be used as a debugging aid when you are formatting expressions in a `Format-List` command, and the expressions do not appear to be working.

-View <System.String>
Specifies the name of an alternate list format or view. You cannot use the Property and View parameters in the same command.

<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: Format computer services -------------

```
Get-Service | Format-List
```

This command formats information about services on the computer as a list. By default, the services are formatted as a table. The `Get-Service` cmdlet gets objects representing the services on the computer. The pipeline operator (`|`) passes the results through the pipeline to `Format-List`. Then, the `Format-List` command formats the service information in a list and sends it to the default output cmdlet for display.

---------------- Example 2: Format PS1XML files ----------------

```
$A = Get-ChildItem $pshome\*.ps1xml
Format-List -InputObject $A
```

The first command gets the objects representing the files and stores them in the `$A` variable.

The second command uses `Format-List` to format information about objects stored in `$A`. This command uses the InputObject parameter to pass the variable to `Format-List`, which then sends the formatted output to the default output cmdlet for display.

--------- Example 3: Format process properties by name ---------

```
Get-Process | Format-List -Property Name, BasePriority, PriorityClass
```

It uses the `Get-Process` cmdlet to get an object representing each process. The pipeline operator (`|`) passes the process objects through the pipeline to `Format-List`. `Format-List` formats the processes as a list of the specified properties. The Property parameter name is optional, so you can omit it.

-------- Example 4: Format all properties for a process --------

```
Get-Process winlogon | Format-List -Property *
```

It uses the Get-Process cmdlet to get an object representing the Winlogon process. The pipeline operator (`|`) passes the Winlogon process object through the pipeline to `Format-List`. The command uses the Property parameter to specify the properties and the `*` to indicate all properties. Because the name of the Property parameter is optional, you can omit it and type the command as `Format-List *`. `Format-List` automatically sends the results to the default output cmdlet for display.

----------- Example 5: Troubleshooting format errors -----------

```
PC /> Get-Date | Format-List DayOfWeek,{ $_ / $null } -DisplayError


DayOfWeek    : Friday
 $_ / $null  : #ERR


PC /> Get-Date | Format-List DayOfWeek,{ $_ / $null } -ShowError


DayOfWeek    : Friday
 $_ / $null  :


Failed to evaluate expression " $_ / $null ".
+ CategoryInfo          : InvalidArgument: (12/21/2018 7:59:23 AM:PSObject)
[], RuntimeException
+ FullyQualifiedErrorId : PSPropertyExpressionError
```

REMARKS

To see the examples, type: "get-help Format-List -examples".

For more information, type: "get-help Format-List -detailed".

For technical information, type: "get-help Format-List -full".

For online help, type: "get-help Format-List -online"