



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 'Format-Custom'

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

NAME

Format-Custom

SYNOPSIS

Uses a customized view to format the output.

SYNTAX

```
Format-Custom [[-Property] <System.Object[]> [-Depth <System.Int32>]
[-DisplayError] [-Expand {CoreOnly | EnumOnly | Both}] [-Force] [-GroupBy
<System.Object>] [-InputObject <System.Management.Automation.PSObject>]
[-ShowError] [-View <System.String>] [<CommonParameters>]
```

DESCRIPTION

The `Format-Custom` cmdlet formats the output of a command as defined in an alternate view. `Format-Custom` is designed to display views that are not just tables or just lists. You can use the views defined in PowerShell, or you can create your own views in a new `format.ps1xml` file and use the `Update-FormatData` cmdlet to add them to PowerShell.

PARAMETERS

-Depth <System.Int32>

Specifies the number of columns in the display.

-DisplayError <System.Management.Automation.SwitchParameter>

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-Custom` command, and the expressions do not appear to be working.

-Expand <System.String>

Formats the 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 .

Valid values 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 objects in the collection.

-Force <System.Management.Automation.SwitchParameter>

Directs the cmdlet to display all of the error information. Use with the DisplayError or ShowError parameters. By default, when an error object is written to the error or display streams, only some of the error information is displayed.

-GroupBy <System.Object>

Formats 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:

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

- Depth - ``<int32>``

For more information, see `about_Calculated_Properties`

(`../Microsoft.PowerShell.Core/About/about_Calculated_Properties.md`).

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

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-Custom`` command, and the expressions do not appear to be working.

`-View <System.String>`

Specifies the name of an alternate format or view. If you omit this parameter, ``Format-Custom`` uses a default custom 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 output with a custom view -----

```
Get-Command Start-Transcript | Format-Custom -View MyView
```

This command formats information about the `Start-Transcript` cmdlet in the format defined by the `MyView` view, a custom view created by the user. To run this command successfully, you must first create a new PS1XML file, define the `MyView` view, and then use the `Update-FormatData` command to add the PS1XML

file to PowerShell.

----- Example 2: Format output with the default view -----

```
Get-Process Winlogon | Format-Custom
```

This command formats information about the Winlogon process in an alternate customized view. Because the command does not use the View parameter, `Format-Custom` uses a default custom view to format the data.

----- Example 3: Troubleshooting format errors -----

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

```
class DateTime
{
    DayOfWeek = Friday
    $_ / $null = #ERR
}
```

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

```
class DateTime
{
    DayOfWeek = Friday
    $_ / $null =
}
```

Failed to evaluate expression " \$_ / \$null ".

+ CategoryInfo : InvalidArgument: (12/21/2018 8:01:04 AM:PSObject)

[], RuntimeException

+ FullyQualifiedErrorId : PSPropertyExpressionError

REMARKS

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

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

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

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