PowerShell Get-Help on command 'Format-Wide'

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

NAME

Format-Wide

SYNOPSIS

Formats objects as a wide table that displays only one property of each object.

SYNTAX

Format-Wide [[-Property] <System.Object>] [-AutoSize] [-Column <System.Int32>]

[-DisplayError] [-Expand {CoreOnly | EnumOnly | Both}] [-Force] [-GroupBy

<System.Object>] [-InputObject <System.Management.Automation.PSObject>]

[-ShowError] [-View <System.String>] [<CommonParameters>]

DESCRIPTION

The `Format-Wide` cmdlet formats objects as a wide table that displays only

one property of each object. You can use the Property parameter to determine

which property is displayed.

PARAMETERS

-AutoSize <System.Management.Automation.SwitchParameter>
   Adjusts the column size and number of columns based on the width of the

   data. By default, the column size and number are determined by the view.

   You cannot use the AutoSize and Column parameters in the same command.

-Column <System.Int32>
   Specifies the number of columns in the display. You cannot use the

   AutoSize and Column parameters in the same command.

-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-Wide` 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

   properties of objects in the

   collection.

-Force <System.Management.Automation.SwitchParameter>
   Indicates that this cmdlet overrides restrictions that prevent the command

   from succeeding, just so the changes do not compromise security. For

example, Force will override the read-only attribute or create directories
to complete a file path, but it will not attempt to change file
permissions.

-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 format. Enter a variable that contains the
objects, or type a command or expression that gets the objects.

-Property <System.Object>

Specifies the object property that appears in the display. 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>`

- `FormatString` - `<string>`

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

-View <System.String>
    Specifies the name of an alternate table 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 names of files in the current directory --

Get-ChildItem | Format-Wide -Column 3

The `Get-ChildItem` cmdlet gets objects representing each file in the directory. The pipeline operator (`` ` | ` ``) passes the file objects through the pipeline to `Format-Wide`, which formats them for output. The Column parameter specifies the number of columns.

----------- Example 2: Format names of registry keys -----------

Get-ChildItem HKCU:\software\microsoft | Format-Wide -Property pschildname -AutoSize

The `Get-ChildItem` cmdlet gets objects representing the keys. The path is specified as `HKCU:`, one of the drives exposed by the PowerShell Registry provider, followed by the key path. The pipeline operator (`` ` | ` ``) passes the registry key objects through the pipeline to `Format-Wide`, which formats them for output. The Property parameter specifies the name of the property, and the AutoSize parameter adjusts the columns for readability.

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

PS /> Get-Date | Format-Wide { $_ / $null } -DisplayError

#ERR

PS /> Get-Date | Format-Wide { $_ / $null } -ShowError

Failed to evaluate expression " $_ / $null ".
+ CategoryInfo   : InvalidArgument: (12/21/2018 8:18:01 AM:PSObject)
[], RuntimeException
+ FullyQualifiedErrorId : PSPropertyExpressionError

REMARKS

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

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

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

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