



Windows PowerShell Get-Help on Cmdlet 'Set-PSDebug'

PS:\>Get-HELP Set-PSDebug -Full

NAME

Set-PSDebug

SYNOPSIS

Turns script debugging features on and off, sets the trace level, and toggles strict mode.

SYNTAX

Set-PSDebug [-Off] [<CommonParameters>]

Set-PSDebug [-Step] [-Strict] [-Trace <System.Int32>] [<CommonParameters>]

DESCRIPTION

The `Set-PSDebug` cmdlet turns script debugging features on and off, sets the trace level, and toggles strict mode. By default, the PowerShell debug features are off.

When the Trace parameter has a value of `1`, each line of script is traced as it runs. When the parameter has a value of `2`, variable assignments, function calls,

and script calls are also traced. If the Step parameter is specified, you're prompted before each line of the script.

PARAMETERS

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

Turns off all script debugging features.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

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

Turns on script stepping. Before each line runs, PowerShell prompts you to stop, continue, or enter a new interpreter level to inspect the state of the script.

Specifying the Step parameter automatically sets a trace level of `1`.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

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

Specifies that variables must be assigned a value before being referenced in a script. If a variable is referenced before a value is assigned, PowerShell returns

an exception error. This is equivalent to ``Set-StrictMode -Version 1``. For more information, see `Set-StrictMode (Set-StrictMode.md)`.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Trace <System.Int32>

Specifies the trace level for each line in a script. Each line is traced as it's run.

The acceptable values for this parameter are as follows:

- 0: Turn script tracing off.

- 1: Trace script lines as they run.

- 2: Trace script lines, variable assignments, function calls, and scripts.

Required? false

Position? named

Default value None

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

None

You can't pipe objects to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

----- Example 1: Set the trace level -----

```
Set-PSDebug -Trace 2; foreach ($i in 1..3) {$i}
```

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ($i in >>>> 1..3) {$i}
```

```
DEBUG: ! SET $foreach = 'IEnumerator'.
```

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ( >>>> $i in 1..3) {$i}
```

```
DEBUG: ! SET $i = '1'.
```

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ($i in 1..3) { >>>> $i}
```

1

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ( >>>> $i in 1..3) {$i}
```

```
DEBUG: ! SET $i = '2'.
```

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ($i in 1..3) { >>>> $i}
```

2

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ( >>>> $i in 1..3) {$i}
```

```
DEBUG: ! SET $i = '3'.
```

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ($i in 1..3) { >>>> $i}
```

3

```
DEBUG: 1+ Set-PSDebug -Trace 2; foreach ( >>>> $i in 1..3) {$i}
```

```
DEBUG: ! SET $foreach = ''.
```

----- Example 2: Turn on stepping -----

```
Set-PSDebug -Step; foreach ($i in 1..3) {$i}
```

Continue with this operation?

```
1+ Set-PSDebug -Step; foreach ($i in >>>> 1..3) {$i}
```

[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): A

```
DEBUG: 1+ Set-PSDebug -Step; foreach ($i in >>>> 1..3) {$i}
```

```
1  
2  
3
```

----- Example 3: Use strict mode -----

```
Set-PSDebug -Strict; $NewVar
```

The variable '\$NewVar' cannot be retrieved because it has not been set.

At line:1 char:22

```
+ Set-PSDebug -Strict; $NewVar
```

----- Example 4: Turn off debug features -----

```
Set-PSDebug -Off; foreach ($i in 1..3) {$i}
```

```
1  
2  
3
```

RELATED LINKS

Online

Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.core/set-psdebug?view=powershell-5.1&WT.mc_id=ps-gethelp

about_Debuggers

Debug-Process

Set-PSBreakpoint

Set-StrictMode

Write-Debug