Windows PowerShell Get-Help on Cmdlet 'Disable-PSBreakpoint'

PS:\>Get-HELP Disable-PSBreakpoint -Full

NAME

Disable-PSBreakpoint

SYNOPSIS

Disables the breakpoints in the current console.

SYNTAX

Disable-PSBreakpoint [-Breakpoint] <System.Management.Automation.Breakpoint[]> [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Disable-PSBreakpoint [-Id] <System.Int32[]> [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Disable-PSBreakpoint` cmdlet disables breakpoints, which assures that they are not hit when the script runs. You can use it to disable all breakpoints, or you

can specify breakpoints by submitting breakpoint objects or breakpoint IDs.

Technically, this cmdlet changes the value of the Enabled property of a breakpoint object to False. To enable a

breakpoint, use the `Enable-PSBreakpoint` cmdlet.

Breakpoints are enabled by default when you create them using the `Set-PSBreakpoint` cmdlet.

A breakpoint is a point in a script where execution stops temporarily so that you can examine the instructions in the script. `Disable-PSBreakpoint` is one of several

cmdlets designed for debugging PowerShell scripts. For more information about the PowerShell debugger, see about_Debuggers

(../microsoft.powershell.core/about/about_debuggers.md).

PARAMETERS

-Breakpoint <System.Management.Automation.Breakpoint[]>

Specifies the breakpoints to disable. Enter a variable that contains breakpoint objects or a command that gets breakpoint objects, such as a `Get-PSBreakpoint`

command. You can also pipe breakpoint objects to the `Disable-PSBreakpoint` cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Id <System.Int32[]>

Disables the breakpoints with the specified breakpoint IDs. Enter the IDs or a variable that contains the IDs. You cannot pipe IDs to `Disable-PSBreakpoint`.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the enabled breakpoints. By default, this cmdlet does not generate any output.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

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.Breakpoint

You can pipe a breakpoint object to this cmdlet.

OUTPUTS

None

By default, this cmdlet returns no output.

System.Management.Automation.Breakpoint

When you use the PassThru parameter, this cmdlet returns a breakpoint object representing the disabled breakpoint.

NOTES

Windows PowerShell includes the following aliases for `Disable-PSBreakpoint`:

- `dbp`

---------- Example 1: Set a breakpoint and disable it ----------

```
$B = Set-PSBreakpoint -Script "sample.ps1" -Variable "name"
$B | Disable-PSBreakpoint
```

The `Set-PSBreakpoint` cmdlet creates a breakpoint on the `$Name` variable in the `Sample.ps1` script and saves the breakpoint object in the `$B` variable. The

`Disable-PSBreakpoint` cmdlet disables the new breakpoint. It uses a pipeline operator (`|`) to send the breakpoint object in `$B` to the `Disable-PSBreakpoint`

cmdlet.

As a result of this command, the value of the Enabled property of the breakpoint object in `$B` is False .

---------- Example 2: Disable a breakpoint ----------

Disable-PSBreakpoint -Id 0

---------- Example 3: Create a disabled breakpoint ----------

Disable-PSBreakpoint -Breakpoint ($B = Set-PSBreakpoint -Script "sample.ps1" -Line 5)

It uses the `Disable-PSBreakpoint` cmdlet to disable the breakpoint. The value of the Breakpoint parameter is a `Set-PSBreakpoint` command that sets a new breakpoint,

generates a breakpoint object, and saves the object in the `$B` variable.

Cmdlet parameters that take objects as their values can accept a variable that contains the object or a command that gets or generates the object. In this case,

because `Set-PSBreakpoint` generates a breakpoint object, it can be used as the value of the Breakpoint parameter.

-- Example 4: Disable all breakpoints in the current console --

`Get-PSBreakpoint` | Disable-PSBreakpoint

RELATED LINKS

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

Enable-PSBreakpoint

Get-PSBreakpoint

Get-PSCallStack

Remove-PSBreakpoint

Set-PSBreakpoint