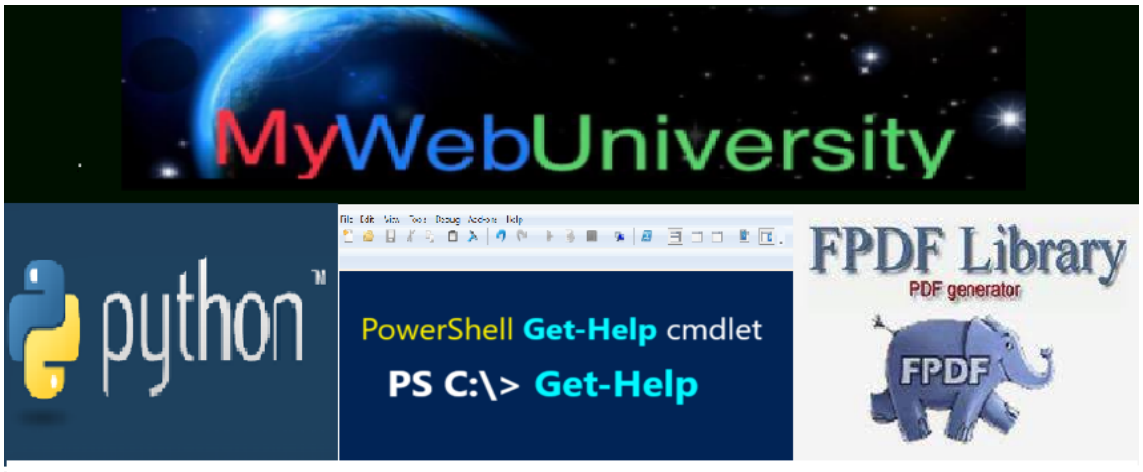

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Debug-Runspace'

PS:\>Get-HELP Debug-Runspace -Full

NAME

Debug-Runspace

SYNOPSIS

Starts an interactive debugging session with a runspace.

SYNTAX

Debug-Runspace [-Id] <System.Int32> [-Confirm] [-WhatIf] [<CommonParameters>]

Debug-Runspace [-InstanceId] <System.Guid> [-Confirm] [-WhatIf] [<CommonParameters>]

Debug-Runspace [-Name] <System.String> [-Confirm] [-WhatIf] [<CommonParameters>]

Debug-Runspace [-Runspace] <System.Management.Automation.Runspaces.Runspace> [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Debug-Runspace` cmdlet starts an interactive debugging session with a local or remote active runspace. You can

find a runspace that you want to debug by first

running `Get-Process` to find processes associated with PowerShell, then `Enter-PSHostProcess` with the process ID specified in the Id parameter to attach to the

process, and then `Get-Runspace` to list runspaces within the PowerShell host process.

After you have selected a runspace to debug, if the runspace is currently running a command or script, or if the script has stopped at a breakpoint, PowerShell opens

a remote debugger session for the runspace. You can debug the runspace script in the same way remote session scripts are debugged.

You can only attach to a PowerShell host process if you are an administrator on the computer that is running the process, or you are running the script that you want

to debug. Also, you cannot enter the host process that is running the current PowerShell session. You can only enter a host process that is running a different

PowerShell session.

PARAMETERS

-Id <System.Int32>

Specifies the ID number of a runspace. You can run `Get-Runspace` to show runspace IDs.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InstanceId <System.Guid>

Specifies a runspace by its instance ID, a GUID that you can show by running `Get-Runspace`.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Name <System.String>

Specifies a runspace by its name. You can run `Get-Runspace` to show the names of runspaces.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Runspace <System.Management.Automation.Runspaces.Runspace>

Specifies a runspace object. The simplest way to provide a value for this parameter is to specify a variable that contains the results of a filtered

`Get-Runspace` command.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value True

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 True

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.Runspaces.Runspace

You can pipe the results of a `Get-Runspace` command to this cmdlet.

OUTPUTS

NOTES

`Debug-Runspace` works on runspaces that are in the Opened state. If a runspace state changes from Opened to another state, that runspace is automatically removed

from the running list. A runspace is added to the running list only if it meets the following criteria.

- If it is coming from Invoke-Command; that is, it has an `Invoke-Command` GUID ID.

- If it is coming from `Debug-Runspace`; that is, it has a `Debug-Runspace` GUID ID.

- If it is coming from a PowerShell workflow, and its workflow job ID is the same as the current

active debugger workflow job ID.

---------- Example 1: Debug a remote runspace ----------

```
PS C:\> Get-Process -ComputerName "WS10TestServer" -Name "*powershell*"

Handles      WS(K)   VM(M)      CPU(s)    Id  ProcessName
-------      -----   -----      ------    --  -----------
    377      69912     63     2.09      2420  powershell
    399     123396    829     4.48      1152  powershell_ise

PS C:\> Enter-PSSession -ComputerName "WS10TestServer"
[WS10TestServer]:PS C:\> Enter-PSHostProcess -Id 1152
[WS10TestServer:][Process:1152]: PS C:\Users\Test\Documents> Get-Runspace

Id Name            ComputerName    Type          State         Availability
-- ----            ------------    ----          -----         ------------
 1 Runspace1       WS10TestServer  Remote        Opened        Available
 2 RemoteHost      WS10TestServer  Remote        Opened        Busy

[WS10TestServer][Process:1152]: PS C:\Users\Test\Documents> Debug-Runspace -Id 2

Hit Line breakpoint on 'C:\TestWFVar1.ps1:83'
At C:\TestWFVar1.ps1:83 char:1
+ $scriptVar = "Script Variable"
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[Process:1152]: [RSDBG: 2]: PS C:\> >
```

In the second command, you run `Enter-PSSession` to open a remote session on WS10TestServer. In the third command, you attach to the Windows PowerShell ISE host

process running on the remote server by running `Enter-PSHostProcess`, and specifying the ID of the host process that you obtained in the first command, 1152.

In the fourth command, you list available runspaces for process ID 1152 by running `Get-Runspace`. You note the ID number of the Busy runspace; it is running a script

that you want to debug.

In the last command, you start debugging an opened runspace that is running a script, `TestWFVar1.ps1`, by running `Debug-Runspace`, and identifying the runspace by

its ID, 2, by adding the Id parameter. Because there's a breakpoint in the script, the debugger opens.

RELATED LINKS

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

about_Debuggers

Debug-Job

Get-Runspace

Get-Process

Enter-PSHostProcess

Enter-PSSession