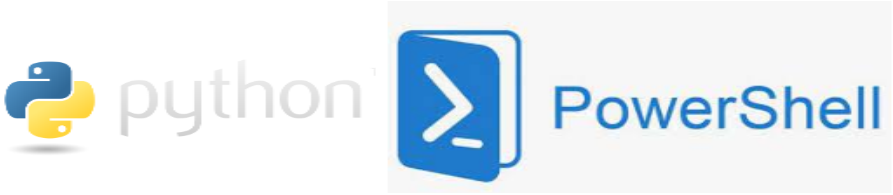

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Debug-Runspace'

PS C:\Users\wahid> Get-Help Debug-Runspace

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.

-InstanceId <System.Guid>

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

-Name <System.String>

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

-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.

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

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

<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: 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.

REMARKS

To see the examples, type: "get-help Debug-Runspace -examples".
For more information, type: "get-help Debug-Runspace -detailed".
For technical information, type: "get-help Debug-Runspace -full".
For online help, type: "get-help Debug-Runspace -online"