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

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

NAME

Debug-Job

SYNOPSIS

Debugs a running background, remote, or Windows PowerShell Workflow job.

SYNTAX

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

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

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

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

DESCRIPTION

The `Debug-Job` cmdlet lets you debug scripts that are running within jobs. The cmdlet is designed to debug PowerShell Workflow jobs, background jobs, and jobs

running in remote sessions. `Debug-Job` accepts a running job object, name, ID, or instance ID as input, and starts a debugging session on the script it is running.

The debugger `quit` command stops the job and running script. The `exit` command detaches the debugger, and allows the job to continue to run.

PARAMETERS

-Id <System.Int32>

Specifies the ID number of a running job. To get the ID number of a job, run the `Get-Job` cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InstanceId <System.Guid>

Specifies the instance ID GUID of a running job.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Job <System.Management.Automation.Job>

Specifies a running job object. The simplest way to use this parameter is to save the results of a `Get-Job` command that returns the running job that you want to

debug in a variable, and then specify the variable as the value of this parameter.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Name <System.String>

Specifies a job by the friendly name of the job. When you start a job, you can specify a job name by adding the JobName parameter, in cmdlets such as

`Invoke-Command` and `Start-Job`.

Required? true

Position? 0

Default value None

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.RemotingJob

OUTPUTS

NOTES

---------- Example 1: Debug a job by job ID ----------

Debug-Job -ID 3

Id	Name	PSJobTypeName	State	HasMoreData	Location	Command
3	Job3	RemoteJob	Running	True	PowerShellIx	TestWFDemo1.ps1

Entering debug mode. Use h or ? for help.

Hit Line breakpoint on 'C:\TestWFDemo1.ps1:8'

At C:\TestWFDemo1.ps1:8 char:5
+ Write-Output -InputObject "Now writing output:"
+ ~~~~~~

[DBG:PowerShellIx]: PS C:\> > list

```
3:
4:  workflow SampleWorkflowTest
5:  {
6:      param ($MyOutput)
7:
8:*     Write-Output -InputObject "Now writing output:"
9:      Write-Output -Input $MyOutput
10:
11:     Write-Output -InputObject "Get PowerShell process:"
12:     Get-Process -Name powershell
13:
14:     Write-Output -InputObject "Workflow function complete."
15:  }
16:
17:  # Call workflow function
18:  SampleWorkflowTest -MyOutput "Hello"
```

RELATED LINKS

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

Get-Job

Receive-Job

Remove-Job

Resume-Job

Start-Job

Stop-Job

Suspend-Job

Wait-Job