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

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

NAME

Disable-ScheduledJob

SYNOPSIS

Disables a scheduled job.

SYNTAX

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

Disable-ScheduledJob [-InputObject] <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Disable-ScheduledJob [-Name] <System.String> [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Disable-ScheduledJob` cmdlet temporarily disables scheduled jobs. Disabling preserves all job properties and does not disable the job triggers, but it prevents

the scheduled jobs from starting automatically when triggered. You can start a disabled scheduled job by using the

`Start-Job` cmdlet or use a disabled scheduled job

as a template.

To disable a scheduled job, the `Disable-ScheduledJob` cmdlet sets the Enabled property of the scheduled job to False. To re-enable the scheduled job, use the

`Enable-ScheduledJob` cmdlet.

`Disable-ScheduledJob` is one of a collection of job scheduling cmdlets in the PSScheduledJob module that is included in Windows PowerShell.

For more information about Scheduled Jobs, see the About topics in the PSScheduledJob module. Import the PSScheduledJob module and then type: `Get-Help

about_Scheduled*` or see about_Scheduled_Jobs (About/about_Scheduled_Jobs.md).

This cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

-Id <System.Int32>

Disables the scheduled job with the specified identification number (ID). Enter the ID of a scheduled job.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition>

Specifies the scheduled job to be disabled. Enter a variable that contains ScheduledJobDefinition objects or type a command or expression that gets

ScheduledJobDefinition objects, such as a `Get-ScheduledJob` command. You can also pipe a ScheduledJobDefinition object to `Disable-ScheduledJob`.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Name <System.String>

Disables the scheduled jobs with the specified names. Enter the name of a scheduled job. Wildcards are supported.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the item with which you are working. 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

Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition

You can pipe a scheduled job to `Disable-ScheduledJob`.

OUTPUTS

None

By default, this cmdlet returns no output.

Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition

When you use the PassThru parameter, this cmdlet returns the scheduled job that is disabled.

NOTES

- `Disable-ScheduledJob` does not generate warnings or errors if you use it to disable a scheduled job that is already

disabled.

---------- Example 1: Disable a scheduled job ----------

Disable-ScheduledJob -ID 2 -Passthru

This command disables the scheduled job with ID 2 on the local computer.

---------- Example 2: Disable all scheduled jobs ----------

Get-ScheduledJob | Disable-ScheduledJob -Passthru

Id	Name	Triggers	Command	Enabled
1	ArchiveProje...	{}	C:\Scripts\Archive-DxProjects.ps1	False
2	Inventory	{1, 2}	\\Srv01\Scripts\Get-FullInventory.ps1	False
4	Test-HelpFiles	{1}	.\Test-HelpFiles.ps1	False
5	TestJob	{1, 2}	.\Run-AllTests.ps1	False

The `Get-ScheduledJob` cmdlet to gets all scheduled job and pipes them to the `Disable-ScheduledJob` cmdlet to disable them.

You can re-enable scheduled job by using the `Enable-ScheduledJob` cmdlet and run a disabled scheduled job by using the `Start-Job` cmdlet.

`Disable-ScheduledJob` does not generate warnings or errors if you disable a scheduled job that is already disabled, so you can disable all scheduled jobs without

conditions.

---------- Example 3: Disable selected scheduled jobs ----------

Get-ScheduledJob | Where-Object {!$_.Credential} | Disable-ScheduledJob

Jobs without credentials run with the permission of the user who created them.

The command uses the `Get-ScheduledJob` cmdlet to get all scheduled jobs on the computer. A pipeline operator sends the scheduled jobs to the `Where-Object` cmdlet,

which selects scheduled jobs that do not have credentials. The command uses the not (`!`) operator and references the Credential property of the scheduled job.

Another pipeline operator sends the selected scheduled jobs to the `Disable-ScheduledJob` cmdlet, which disables them.

---- Example 4: Disable scheduled jobs on a remote computer ----

Invoke-Command -ComputerName Srv01, Srv10 -ScriptBlock {Disable-ScheduledJob -Name TestJob}

The command uses the `Invoke-Command` cmdlet to run a `Disable-ScheduledJob` command on the Srv01 and Srv10 computers. The command uses the Name parameter of

`Disable-ScheduledJob` to select the TestJob scheduled job on each computer.

----- Example 5: Disable a scheduled job by its global ID -----

Get-ScheduledJob | Format-Table -Property Name, GlobalID, Command -Autosize

Name GlobalId Command

---- -------- -------

ArchiveProjects1 a26a0b3d-b4e6-44d3-8b95-8706ef621f7c C:\Scripts\Archive-DxProjects.ps1

Inventory 3ac37e5d-84c0-4a8f-9661-7e88ebb8f914 \\Srv01\Scripts\Get-FullInventory.ps1

Backup-Scripts 4d0cc6be-c082-48d1-baec-1bd8278f3c81 Copy-Item C:\CurrentScripts*.ps1 -Destination C:\BackupScripts

Test-HelpFiles d77020ca-f20d-42be-86c8-fc64df97db90 .\Test-HelpFiles.ps1

Test-HelpFiles 2f1606d2-c6cf-4bef-8b1c-ae36a9cc9934 .\Test-DomainHelpFiles.ps1

Get-ScheduledJob | Where-Object {$_.GlobalID = d77020ca-f20d-42be-86c8-fc64df97db90} | Disable-ScheduledJob

The first command demonstrates one way of finding the GlobalID of a scheduled job. The command uses the `Get-ScheduledJob` cmdlet to get the scheduled jobs on the

computer. A pipeline operator (`|`) sends the scheduled jobs to the `Format-Table` cmdlet, which displays the Name, GlobalID, and Command properties of each job in a

table.

The second command uses the `Get-ScheduledJob` cmdlet to get the scheduled jobs on the computer. A pipeline operator (`|`) sends the scheduled jobs to the

`Where-Object` cmdlet, which selects the scheduled job with the specified global ID. Another pipeline operator sends the job to the `Disable-ScheduledJob` cmdlet,

which disables it.

RELATED LINKS

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

Add-JobTrigger

Disable-JobTrigger

Disable-ScheduledJob

Enable-JobTrigger

Enable-ScheduledJob

Get-JobTrigger

Get-ScheduledJob

Get-ScheduledJobOption

New-JobTrigger

New-ScheduledJobOption

Register-ScheduledJob

Remove-JobTrigger

Set-JobTrigger

Set-ScheduledJob

Set-ScheduledJobOption

Unregister-ScheduledJob