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 'Set-ScheduledJob'

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

```
NAME

    Set-ScheduledJob


SYNOPSIS

    Changes scheduled jobs.



SYNTAX

        Set-ScheduledJob [-InputObject] <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [-ArgumentList
<System.Object[]>] [-Authentication {Default | Basic |

            Negotiate | NegotiateWithImplicitCredential | Credssp | Digest | Kerberos}] [-Credential
<System.Management.Automation.PSCredential>] [-InitializationScript

        <System.Management.Automation.ScriptBlock>] [-MaxResultCount <System.Int32>] [-Name <System.String>]
[-PassThru] [-RunAs32] [-RunEvery <System.TimeSpan>] [-RunNow]

               [-ScheduledJobOption <Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions>] [-ScriptBlock
<System.Management.Automation.ScriptBlock>] [-Trigger

    <Microsoft.PowerShell.ScheduledJob.ScheduledJobTrigger[]>] [<CommonParameters>]


        Set-ScheduledJob [-InputObject] <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [-ArgumentList
<System.Object[]>] [-Authentication {Default | Basic |
```

Negotiate | NegotiateWithImplicitCredential | Credssp | Digest | Kerberos}] [-Credential <System.Management.Automation.PSCredential>] [-FilePath <System.String>]

[-InitializationScript <System.Management.Automation.ScriptBlock>] [-MaxResultCount <System.Int32>] [-Name <System.String>] [-PassThru] [-RunAs32] [-RunEvery

<System.TimeSpan>] [-RunNow] [-ScheduledJobOption <Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions>] [-Trigger

<Microsoft.PowerShell.ScheduledJob.ScheduledJobTrigger[]>] [<CommonParameters>]

Set-ScheduledJob [-InputObject] <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [-ClearExecutionHistory] [-PassThru] [<CommonParameters>]

DESCRIPTION

The `Set-ScheduledJob` cmdlet changes the properties of scheduled jobs, such as the commands that the jobs run or the credentials required to run the job. You can

also use it to clear the execution history of the scheduled job.

To use this cmdlet, begin by using the `Get-ScheduledJob` cmdlet to get the scheduled job. Then, pipe the scheduled job to `Set-ScheduledJob` or save the job in a

variable and use the InputObject parameter to identify the job. Use the remaining parameters of `Set-ScheduledJob` to change the job properties or clear the execution

history.

Although you can use `Set-ScheduledJob` to change the triggers and options of a scheduled job, the `Add-JobTrigger`, `Set-JobTrigger`, and `Set-ScheduledJobOption`

cmdlets provide much easier ways to accomplish those tasks. To create a new scheduled job, use the `Register-ScheduledJob` cmdlet.

The Trigger parameter of `Set-ScheduledJob` adds one or more job triggers that start the job. The Trigger parameter is optional, so you can add triggers when you

create the scheduled job, add job triggers later, add the RunNow parameter to start the job immediately, use the `Start-Job` cmdlet to start the job immediately at

any time, or save the untriggered scheduled job as a template for other jobs.

`Set-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

-ArgumentList <System.Object[]>

Specifies values for the parameters of the script that is specified by the FilePath parameter or for the command that is specified by the ScriptBlock parameter.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-Authentication <System.Management.Automation.Runspaces.AuthenticationMechanism>

Specifies the mechanism that is used to authenticate the user's credentials. The acceptable values for this parameter are:

- `Default`

- `Basic`

- `Credssp`

- `Digest`

- `Kerberos`

- `Negotiate`

- `NegotiateWithImplicitCredential`

The default value is `Default`. For more information about the values of this parameter, see AuthenticationMechanism Enumeration

(/dotnet/api/system.management.automation.runspaces.authenticationmechanism)in the PowerShell SDK.

> [!CAUTION] > Credential Security Support Provider (CredSSP) authentication, in which the user's credentials are > passed to a remote computer to be

authenticated, is designed for commands that require > authentication on more than one resource, such as accessing a remote network share. This mechanism >

increases the security risk of the remote operation. If the remote computer is compromised, the > credentials that are passed to it can be used to control the

network session.

```
Required?                    false
Position?                    named
Default value                None
Accept pipeline input?       False
Accept wildcard characters?  false
```

-ClearExecutionHistory <System.Management.Automation.SwitchParameter>

Deletes the current execution history and the saved results of the scheduled job.

The job execution history and job results are saved with the scheduled job in the `$HOME\AppData\Local\Microsoft\Windows\PowerShell\ScheduledJobs` directory on

the computer on which the job is created. To see the execution history, use the `Get-Job` cmdlet. To get the job results, use the `Receive-Job` cmdlet.

This parameter does not affect the events that Task Scheduler writes to the Windows event logs and it does not stop Windows PowerShell from saving job results. To manage the number of job results that are saved, use the MaxResultCount parameter.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to run the scheduled job. The default is the current user.

Type a user name, such as User01 or Domain01\User01, or enter a PSCredential object, such as one from the `Get-Credential` cmdlet. If you enter only a user name, you will be prompted for a password.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-FilePath <System.String>

Specifies a script that the scheduled job runs. Enter the path to a .ps1 file on the local computer. To specify default values for the script parameters, use the ArgumentList parameter. Every scheduled job must have either a ScriptBlock or FilePath value.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InitializationScript <System.Management.Automation.ScriptBlock>

Specifies the fully qualified path to a Windows PowerShell script (`.ps1`). The initialization script runs in the session that is created for the background job

before the commands that are specified by the ScriptBlock parameter or the script that is specified by the FilePath parameter. You can use the initialization

script to configure the session, such as adding files, functions, or aliases, creating directories, or checking for prerequisites.

To specify a script that runs the primary job commands, use the FilePath parameter.

If the initialization script generates an error, including a non-terminating error, the current instance of the scheduled job does not run and its status is

Failed.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-InputObject <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition>

Specifies the scheduled job to be changed. 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 `Set-ScheduledJob`.

If you specify multiple scheduled jobs, `Set-ScheduledJob` makes the same changes to all jobs.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByValue)

Accept wildcard characters? false

-MaxResultCount <System.Int32>

Specifies how many job result entries are maintained for the scheduled job. The default value is 32.

Windows PowerShell saves the execution history and results of each triggered instance of the scheduled job on disk. The value of this parameter determines the

number of job instance results that are saved for this scheduled job. When the number of job instance results exceeds this value, Windows PowerShell deletes the

results of the oldest job instance to make room for the results of the newest job instance.

The job execution history and job results are saved in the `$HOME\AppData\Local\Microsoft\Windows\PowerShell\ScheduledJobs<JobName>\Output<Timestamp>` directories

on the computer on which the job is created. To see the execution history, use the `Get-Job` cmdlet. To get the job results, use the `Receive-Job` cmdlet.

The MaxResultCount parameter sets the value of the ExecutionHistoryLength property of the scheduled job.

To delete the current execution history and job results, use the ClearExecutionHistory parameter.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-Name <System.String>

Specifies a new name for the scheduled job and instances of the scheduled job. The name must be unique on the local computer.

To identify the scheduled job to be changed, use the InputObject parameter or pipe a scheduled job from `Get-ScheduledJob` to `Set-ScheduledJob`.

This parameter does not change the names of job instances on disk. It affects only job instances that are started after this command completes.

Required?   false

Position?   named

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

-RunAs32 <System.Management.Automation.SwitchParameter>

Runs the scheduled job in a 32-bit process.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-RunEvery <System.TimeSpan>

Used to specify how often to run the job. For example, use this option to run a job every 15 minutes.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-RunNow <System.Management.Automation.SwitchParameter>

Starts a job immediately, as soon as the `Set-ScheduledJob` cmdlet is run. This parameter eliminates the need to trigger Task Scheduler to run a Windows

PowerShell script immediately after registration, and does not require users to create a trigger that specifies a starting date and time.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-ScheduledJobOption <Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions>

Sets options for the scheduled job. Enter a ScheduledJobOptions object, such as one that you create by using the `New-ScheduledJobOption` cmdlet, or a hash table

value.

You can set options for a scheduled job when you register the scheduled job or use the `Set-ScheduledJobOption` or `Set-ScheduledJob` cmdlets to set or change

options.

Many of the options and their default values determine whether and when a scheduled job runs. Be sure to review these options before scheduling a job. For a

description of the scheduled job options, including the default values, see `New-ScheduledJobOption`.

To submit a hash table, use the following keys. In the following hash table, the keys are shown with their default values.

`@{# Power SettingsStartIfOnBattery=$False;StopIfGoingOnBattery=$True; WakeToRun=$False; # Idle SettingsStartIfNotIdle=$False; IdleDuration="00:10:00";

IdleTimeout="01:00:00"; StopIfGoingOffIdle=$True; RestartOnIdleResume=$False;# Security settingsShowInTaskScheduler=$TrueRunElevated=$False;#

MiscRunWithoutNetwork=$False;DoNotAllowDemandStart=$False;MultipleInstancePolicy=IgnoreNew# Can be IgnoreNew, Parallel, Queue, StopExisting}`

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-ScriptBlock <System.Management.Automation.ScriptBlock>

Specifies the commands that the scheduled job runs. Enclose the commands in braces (`{}`) to create a script block. To specify default values for command

parameters, use the ArgumentList parameter.

Every `Register-ScheduledJob` command must use either the ScriptBlock or FilePath parameters.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-Trigger <Microsoft.PowerShell.ScheduledJob.ScheduledJobTrigger[]>

Specifies the triggers for the scheduled job. Enter one or more ScheduledJobTrigger objects, such as the objects that the `New-JobTrigger` cmdlet returns, or a

hash table of job trigger keys and values.

A job trigger starts a scheduled job automatically on a one-time or recurring scheduled or when an event occurs.

Job triggers are optional. You can add a trigger when you create the scheduled job, use the `Add-JobTrigger` or `Set-ScheduledJob` cmdlets to add triggers later,

or use the `Start-Job` cmdlet to start the scheduled job immediately. You can also create and maintain a scheduled job that has no job triggers.

To submit a hash table, use the following keys.

`@{Frequency="Once" (or Daily, Weekly, AtStartup, AtLogon);At="3am"` (or any valid time string); `DaysOfWeek="Monday", "Wednesday"` (or any combination of day

names); `Interval=2` (or any valid frequency interval); `RandomDelay="30minutes"` (or any valid timespan string); `User="Domain1\User01"` (or any valid user; used

only with the AtLogon frequency value)

}

```
Required?                    false
Position?                    named
Default value                None
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 this cmdlet.

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 it changed.

NOTES

--------- Example 1: Change the script that a job runs ---------

```
Get-ScheduledJob -Name "Inventory"

Id         Name            Triggers       Command                              Enabled
--        ----           --------       -------                              -------
1          Inventory       {1}            C:\Scripts\Get-Inventory.ps1             True

Get-ScheduledJob -Name "Inventory" | Set-ScheduledJob -FilePath "C:\Scripts\Get-FullInventory.ps1" -Passthru

Id         Name            Triggers       Command                              Enabled
--        ----           --------       -------                              -------
1          Inventory       {1}            C:\Scripts\Get-FullInventory.ps1         True
```

The first command uses the `Get-ScheduledJob` cmdlet to get the Inventory scheduled job. The output shows that the job runs the Get-Inventory.ps1 script.

The second command uses the `Get-ScheduledJob` cmdlet to get the Inventory scheduled job. A pipeline operator (`|`) sends the scheduled job to the `Set-ScheduledJob`

cmdlet. The `Set-ScheduledJob` cmdlet uses the Script parameter to specify a new script, `Get-FullInventory.ps1`. The command uses the Passthru parameter to return the scheduled job after the change.

This command is not required; it is included only to show the effect of the script change.

-- Example 2: Delete the execution history of a scheduled job --

```
Get-ScheduledJob BackupArchive | Set-ScheduledJob -ClearExecutionHistory
```

The command uses the `Get-ScheduledJob` cmdlet to get the BackupArchive scheduled job. A pipeline operator (`|`) sends the job to the `Set-ScheduledJob` cmdlet to

change it. The `Set-ScheduledJob` cmdlet uses the ClearExecutionHistory parameter to delete the execution history and saved results.

For more information about the execution history and saved job results of scheduled jobs, see about_Scheduled_Jobs (About/about_Scheduled_Jobs.md).

---- Example 3: Change scheduled jobs on a remote computer ----

```
Invoke-Command -Computer "Server01, Server02" -ScriptBlock {Get-ScheduledJob |
    Set-ScheduledJob -InitializationScript \\SrvA\Scripts\SetForRun.ps1}
```

The command uses the `Invoke-Command` cmdlet to run a command on the Server01 and Server02 computers.

The remote command begins with a `Get-ScheduledJob` command that gets all scheduled jobs on the computer. The scheduled jobs are piped to the `Set-ScheduledJob`

cmdlet, which changes the initialization script to `SetForRun.ps1`.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/psscheduledjob/set-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