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

PowerShell Get-Help on command 'Set-ScheduledJobOption'

PS C:\Users\wahid> Get-Help Set-ScheduledJobOption

NAME

Set-ScheduledJobOption

SYNOPSIS

Changes the job options of a scheduled job.

SYNTAX

Set-ScheduledJobOption [-InputObject]

<Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions>

[-ContinueIfGoingOnBattery] [-DoNotAllowDemandStart] [-HideInTaskScheduler]

[-IdleDuration <System.TimeSpan>] [-IdleTimeout <System.TimeSpan>]

[-MultipleInstancePolicy {None | IgnoreNew | Parallel | Queue | StopExisting}]

[-PassThru] [-RequireNetwork] [-RestartOnIdleResume] [-RunElevated]

[-StartIfIdle] [-StartIfOnBattery] [-StopIfGoingOffIdle] [-WakeToRun]

[<CommonParameters>]

DESCRIPTION

The `Set-ScheduledJobOptions` cmdlet changes the job options of scheduled jobs.

To change the options of a scheduled job, begin by using the `Get-ScheduledJobOption` cmdlet to get the job options of a scheduled job. Then, pipe the options to `Set-ScheduledJobOption` or save the options in a variable and use the InputObject parameter of `Set-ScheduledJobOption` cmdlet to identify the options. Use the remaining parameters of `Set-ScheduledJobOption` to change the job options.

To turn on a job option, use the parameter that sets that option. To turn off an option, type the parameter name, a colon (`:`), and `$false`. For example, to turn off the RunElevated option, type `-RunElevated:$false`.

Each job options object includes a JobDefinition property that contains the scheduled job, so the association with the scheduled job is retained when the job options are changed.

The scheduled job options determine how the job runs when it is started by Task Scheduler. These options to not apply when you use the `Start-Job` cmdlet to start a scheduled job.

`Set-ScheduledJobOption` 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

-ContinueIfGoingOnBattery <System.Management.Automation.SwitchParameter>

Do not stop the scheduled job if the computer switches to battery power

(disconnects from AC power) while the job is running. By default, scheduled jobs stop when the computer disconnects from AC power.

The ContinueIfGoingOnBattery parameter sets the value of the StopIfGoingOnBatteries property of scheduled jobs to `$true`.

-DoNotAllowDemandStart <System.Management.Automation.SwitchParameter>
Start the job only when it is triggered. Users cannot start the job manually, such as by using the Run feature in Task Scheduler.

This parameter only affects Task Scheduler. It does not prevents users from using the `Start-Job` cmdlet to start the job.

The DoNotAllowDemandStart parameter sets the value of the DoNotAllowDemandStart property of scheduled jobs to `$true`.

-HideInTaskScheduler <System.Management.Automation.SwitchParameter>
Do not display the job in Task Scheduler. This value affects only the computer on which the job runs. By default, scheduled tasks appear in Task Scheduler.

Even if a task is hidden, users can display the task by selecting the Show hidden tasks view option in Task Scheduler.

The HideInTaskScheduler parameter sets the value of the ShowInTaskScheduler property of scheduled jobs to `$false`.

-IdleDuration <System.TimeSpan>
Specifies how long the computer must be idle before the job starts. The default value is 10 minutes. If the computer is not idle for the specified duration before the value of IdleTimeout expires, the scheduled job does not run until the next scheduled time, if any.

Enter a timespan object, such as one generated by the `New-TimeSpan` cmdlet, or enter a value in `<hours>:<minutes>:<seconds>` format that is automatically converted to a TimeSpan object.

To enable this value, use the StartIfIdle parameter. By default, the StartIfNotIdle property of scheduled jobs is set to `$true` and Windows PowerShell ignores the IdleDuration and IdleTimeout values.

-IdleTimeout <System.TimeSpan>

Specifies how long the computer must be idle before the job starts. The default value is 10 minutes. If the computer is not idle for the specified duration before the value of IdleTimeout expires, the scheduled job does not run until the next scheduled time, if any.

Enter a timespan object, such as one generated by the `New-TimeSpan` cmdlet, or enter a value in `<hours>:<minutes>:<seconds>` format that is automatically converted to a TimeSpan object.

To enable this value, use the StartIfIdle parameter. By default, the StartIfNotIdle property of scheduled jobs is set to $True and Windows PowerShell ignores the IdleDuration and IdleTimeout values.

-InputObject <Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions>

Specifies the job options. Enter a variable that contains ScheduledJobOptions objects or type a command or expression that gets ScheduledJobOptions objects, such as a `Get-ScheduledJobOption` command. You can also pipe a ScheduledJobOptions object to `Set-ScheduledJobOption`.

-MultipleInstancePolicy
<Microsoft.PowerShell.ScheduledJob.TaskMultipleInstancePolicy>

Determines how the system responds to a request to start an instance of a scheduled job while another instance of the job is running. The acceptable values for this parameter are:

- `IgnoreNew` - The new job instance is ignored. This is the default value.

- `Parallel` - The new job instance starts immediately.

- `Queue` - The new job instance starts as soon as the current instance completes.

- `StopExisting` - The current instance of the job stop and the new instance starts.

To run the job, all conditions for the job schedule must be met. For example, if the conditions that are set by the RequireNetwork , IdleDuration , and IdleTimeout parameters are not satisfied, the job instance is not started, regardless of the value of this parameter.

-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.

-RequireNetwork <System.Management.Automation.SwitchParameter>
Runs the scheduled job only when network connections are available.

If you specify this parameter and the network is not available at the scheduled start time, the job does not run until the next scheduled start time, if any.

The RequireNetwork parameter sets the value of the RunWithoutNetwork property of scheduled jobs to `$false`.

-RestartOnIdleResume <System.Management.Automation.SwitchParameter>

Restarts a scheduled job when the computer becomes idle. This parameter works with the StopIfGoingOffIdle parameter, which suspends a running scheduled job if the computer becomes active (leaves the idle state).

The RestartOnIdleResume parameter sets the value of the RestartOnIdleResume property of scheduled jobs to `$true`.

-RunElevated <System.Management.Automation.SwitchParameter>
Runs the scheduled job with the permissions of a member of the Administrators group on the computer on which the job runs.

To enable a scheduled job to run with Administrator permissions, use the Credential parameter of `Register-ScheduledJob` to provide explicit credential for the job.

The RunElevated parameter sets the value of the RunElevated property of scheduled jobs to `$true`.

-StartIfIdle <System.Management.Automation.SwitchParameter>
Starts the scheduled job if the computer has been idle for the time specified by the IdleDuration parameter before the time specified by the IdleTimeout parameter expires.

By default, the IdleDuration and IdleTimeout parameters are ignored and the job starts at the scheduled start time even if the computer is busy.

If you specify this parameter and the computer is busy (not idle) at the scheduled start time, the job does not run until the next scheduled start time, if any.

The StartIfIdle parameter sets the value of the StartIfNotIdle property of scheduled jobs to `$false`.

-StartIfOnBattery <System.Management.Automation.SwitchParameter>
    Starts the scheduled job even if the computer is running on batteries at
    the scheduled start time. The default value is `$false`.

    The StartIfOnBattery parameter sets the value of the StartIfOnBatteries
    property of scheduled jobs to `$true`.

-StopIfGoingOffIdle <System.Management.Automation.SwitchParameter>
    Suspends a running scheduled job if the computer becomes active (not idle)
    while the job is running.

    By default, a scheduled job that is suspended when the computer becomes
    active resumes when the computer becomes idle again. To change this
    default behavior, use the RestartOnIdleResume parameter.

    The StopIfGoingOffIdle parameter sets the value of the StopIfGoingOffIdle
    property of scheduled jobs to `$true`.

-WakeToRun <System.Management.Automation.SwitchParameter>
    Wakes the computer from a Hibernate or Sleep state at the scheduled start
    time so it can run the job. By default, if the computer is in a Hibernate
    or Sleep state at the scheduled start time, the job does not run.

    The WakeToRun parameter sets the value of the WakeToRun property of
    scheduled jobs to `$true`.

<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: Change job options ----------

```
Get-ScheduledJobOption -Name "DeployPackage"

StartIfOnBatteries     : False
StopIfGoingOnBatteries : True
WakeToRun              : False
StartIfNotIdle         : True
StopIfGoingOffIdle     : False
RestartOnIdleResume    : False
IdleDuration           : 00:10:00
IdleTimeout            : 01:00:00
ShowInTaskScheduler    : True
RunElevated            : False
RunWithoutNetwork      : False
DoNotAllowDemandStart  : False
MultipleInstancePolicy : IgnoreNew
JobDefinition          :

Get-ScheduledJobOption -Name "DeployPackage" |
    Set-ScheduledJobOption -WakeToRun -RequireNetwork:$false -Passthru

StartIfOnBatteries     : False
StopIfGoingOnBatteries : True
WakeToRun              : True
StartIfNotIdle         : True
StopIfGoingOffIdle     : False
RestartOnIdleResume    : False
IdleDuration           : 00:10:00
IdleTimeout            : 01:00:00
ShowInTaskScheduler    : True
RunElevated            : False
RunWithoutNetwork      : True
DoNotAllowDemandStart  : False
```

MultipleInstancePolicy : IgnoreNewJobDefinition :

This example shows how to change the options of a scheduled job on the local computer.

The first command uses the `Get-ScheduledJobOption` cmdlet to get the job options of the DeployPackage scheduled job. The output shows that the WakeToRun and RunElevated properties are set to `$false`.

The second command uses the `Set-ScheduledJobOpton` cmdlet to change the job options so the values of the WakeToRun and RunWithoutNetwork properties are $True. The command uses the Passthru parameter to return the trigger after the change.

This command is not required; it is included only to show the effect of the option change.
--- Example 2: Change an option on all remote scheduled jobs ---

```
Invoke-Command -Computer "Server01" -ScriptBlock {
    Get-ScheduledJob |
        Get-ScheduledJobOption |
        Set-ScheduledJobOption -IdleTimeout 2:00:00
}
```

This command changes the value of the IdleTimeout from one hour (the default value) to two hours on all scheduled jobs on the Server01 computer.

The command uses the `Invoke-Command` cmdlet to run a command on the Server01 computer.

The remote command begins with a `Get-ScheduledJob` command that gets all scheduled jobs on the computer. The scheduled jobs are piped to the `Get-ScheduledJobOption` cmdlet, which gets the job options of the scheduled

jobs. Each job options object contains a JobDefinition property that contains the scheduled job, so the options object remains associated with the scheduled job even when it is changed.

The job triggers are piped to the `Set-ScheduledJobOption` cmdlet, which changes the value of the IdleTimeout option to two hours (2:00:00).

REMARKS

To see the examples, type: "get-help Set-ScheduledJobOption -examples".

For more information, type: "get-help Set-ScheduledJobOption -detailed".

For technical information, type: "get-help Set-ScheduledJobOption -full".

For online help, type: "get-help Set-ScheduledJobOption -online"