PowerShell Get-Help on command 'New-ScheduledJobOption'

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

NAME

New-ScheduledJobOption

SYNOPSIS

Creates an object that contains advanced options for a scheduled job.

SYNTAX

New-ScheduledJobOption [-ContinueIfGoingOnBattery] [-DoNotAllowDemandStart] [-HideInTaskScheduler] [-IdleDuration <System.TimeSpan>] [-IdleTimeout <System.TimeSpan>] [-MultipleInstancePolicy {None | IgnoreNew | Parallel | Queue | StopExisting}] [-RequireNetwork] [-RestartOnIdleResume] [-RunElevated] [-StartIfIdle] [-StartIfOnBattery] [-StopIfGoingOffIdle] [-WakeToRun] [<CommonParameters>]

DESCRIPTION

The `New-ScheduledJobOption` cmdlet creates an object that contains advanced options for a scheduled job.

You can use the ScheduledJobOptions object that `New-ScheduledJobOption` returns to set job options for a new or existing scheduled job. Alternatively, you can set job options by using the `Get-ScheduledJobOption` cmdlet to get the job options of an existing scheduled job or by using a hash table value to represent the job options.

Without parameters, `New-ScheduledJobOption` generates an object that contains the default values for all of the options. Because all of the properties except for the JobDefinition property can be edited, you can use the resulting object as a template, and create standard option objects for your enterprise.

When creating scheduled jobs and setting scheduled job options, review the default values of all scheduled job options. Scheduled jobs run only when all conditions set for their execution are satisfied.

`New-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 scheduled job waits for the computer to be idle. If this timeout expires before the computer remains idle for the time period that is specified by the IdleDuration parameter, the job does not run until the next scheduled time, if any. The default value is one hour.

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.

-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 default value is `IgnoreNew`. The acceptable values for this parameter are:

- `IgnoreNew` - The new job instance is ignored.

- `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 stops 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.

-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: Create a scheduled job option object with default values

```
New-ScheduledJobOption
```

Example 2: Create a scheduled job option object with custom values

```
New-ScheduledJobOption -RequireNetwork -StartIfOnBattery

StartIfOnBatteries     : True
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 : Ignore
NewJobDefinition       :
```

The following command creates a scheduled job object that requires the network and runs the scheduled job even if the computer is not connected to AC power.

The output shows that the RequireNetwork parameter changed the value of the RunWithoutNetwork property to `$false` and the StartIfOnBattery parameter changed the value of the StartIfOnBatteries property to `$true`.

-------- Example 3: Set options for a new scheduled job --------

```
$runAsAdmin = New-ScheduledJobOption -RunElevated
Register-ScheduledJob -Name Backup -FilePath D:\Scripts\Backup.ps1 -Trigger
$Mondays -ScheduledJobOption $RunAsAdmin
Get-ScheduledJobOption -Name Backup

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

```
MultipleInstancePolicy : IgnoreNew
JobDefinition          :
Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition
```

The first command creates a ScheduledJobOptions object with the RunElevated parameter. It saves the object in the `$runAsAdmin` variable.

The second command uses the `Register-ScheduledJob` cmdlet to create a new scheduled job. The value of the ScheduledJobOption parameter is the option object in the value of the `$runAsAdmin` variable.

The third command uses the `Get-ScheduledJobOption` cmdlet to get the job options of the Backup scheduled job.The cmdlet output shows that the RunElevated property is set to `$true` and the JobDefinition property of the job option object is now populated with the scheduled job object for the Backup scheduled job.

Example 4: Sort the properties of a scheduled job option object

```
$options = New-ScheduledJobOption -WakeToRun
$options.PSObject.Properties | Sort-Object -Property Name | Format-Table
-Property Name, Value -Autosize

Name                   Value
----                   -----
DoNotAllowDemandStart  False
IdleDuration           00:10:00
IdleTimeout            01:00:00
JobDefinition
MultipleInstancePolicy IgnoreNew
RestartOnIdleResume    False
RunElevated            False
RunWithoutNetwork      True
ShowInTaskScheduler    True
```

```
    StartIfNotIdle              True
    StartIfOnBatteries          False
    StopIfGoingOffIdle          False
    StopIfGoingOnBatteries      True
    WakeToRun                   True
```

The first command uses the `New-ScheduledJobOption` cmdlet to create a ScheduledJobOptions object. The command uses the WakeToRun parameter and saves the resulting object in the `$options` variable.

To get the properties of $Options as objects, the second command uses the PSObject property of all Windows PowerShell objects and its Properties property. The command then pipes the property objects to the `Sort-Object` cmdlet, which sorts the properties in alphabetical order by name, and then to the `Format-Table` cmdlet, which displays the names and values of the properties in a table.

This format makes it much easier to find the WakeToRun property of the ScheduledJobOptions object in `$options` and to verify that its value was changed from `$false` to `$true`.

REMARKS

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

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

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

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