



python



PowerShell

FPDF Library
PDF generator

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

PowerShell Get-Help on command 'New-ScheduledTaskSettingsSet'

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

NAME

New-ScheduledTaskSettingsSet

SYNOPSIS

Creates a new scheduled task settings object.

SYNTAX

```
New-ScheduledTaskSettingsSet [-AllowStartIfOnBatteries] [-AsJob] [-CimSession  
<CimSession[]>] [-Compatibility {At | V1 | Vista | Win7 | Win8}]  
[-DeleteExpiredTaskAfter <TimeSpan>] [-Disable] [-DisallowDemandStart]  
[-DisallowHardTerminate] [-DisallowStartOnRemoteAppSession]  
[-DontStopIfGoingOnBatteries] [-DontStopOnIdleEnd] [-ExecutionTimeLimit  
<TimeSpan>] [-Hidden] [-IdleDuration <TimeSpan>] [-IdleWaitTimeout <TimeSpan>]  
[-MaintenanceDeadline <TimeSpan>] [-MaintenanceExclusive] [-MaintenancePeriod  
<TimeSpan>] [-MultipleInstances {Parallel | Queue | IgnoreNew}] [-NetworkId  
<String>] [-NetworkName <String>] [-Priority <Int32>] [-RestartCount <Int32>]  
[-RestartInterval <TimeSpan>] [-RestartOnIdle] [-RunOnlyIfIdle]  
[-RunOnlyIfNetworkAvailable] [-StartWhenAvailable] [-ThrottleLimit <Int32>]  
[-WakeToRun] [<CommonParameters>]
```

DESCRIPTION

The `New-ScheduledTaskSettingsSet` cmdlet creates an object that contains scheduled task settings. Each scheduled task has one set of task settings. Use this cmdlet to configure options to manage the behavior of the task upon completion, to manage the behavior of the task if a problem occurs, or to manage the behavior of the task if an instance of the task is already running.

You can use the scheduled task settings to register a new scheduled task or update an existing task registration.

PARAMETERS

`-AllowStartIfOnBatteries` [`<SwitchParameter>`]

Indicates that Task Scheduler starts if the computer is running on battery power.

`-AsJob` [`<SwitchParameter>`]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete.

`-CimSession` `<CimSession[]>`

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a `New-CimSession` (<https://go.microsoft.com/fwlink/p/?LinkId=227967>) or `[Get-CimSession](https://go.microsoft.com/fwlink/p/?LinkId=227966)cmdlet`.

The default is the current session on the local computer.

`-Compatibility` `<CompatibilityEnum>`

Indicates which version of Task Scheduler with which a task is compatible.

The acceptable values for this parameter are:

- At

- V1

- Vista

- Win7

- Win8

-DeleteExpiredTaskAfter <TimeSpan>

Specifies the amount of time that Task Scheduler waits before deleting the task after it expires.

-Disable [<SwitchParameter>]

Indicates that the task is disabled.

-DisallowDemandStart [<SwitchParameter>]

Indicates that the task cannot be started by using either the Run command or the Context menu.

-DisallowHardTerminate [<SwitchParameter>]

Indicates that the task cannot be terminated by using TerminateProcess.

-DisallowStartOnRemoteAppSession [<SwitchParameter>]

Indicates that the task does not start if the task is triggered to run in a Remote Applications Integrated Locally (RAIL) session.

-DontStopIfGoingOnBatteries [<SwitchParameter>]

Indicates that the task does not stop if the computer switches to battery power.

-DontStopOnIdleEnd [<SwitchParameter>]

Indicates that Task Scheduler does not terminate the task if the idle

condition ends before the task is completed.

-ExecutionTimeLimit <TimeSpan>

Specifies the amount of time that Task Scheduler is allowed to complete the task.

-Hidden [<SwitchParameter>]

Indicates that the task is hidden in the Task Scheduler UI.

-IdleDuration <TimeSpan>

Specifies the amount of time that the computer must be in an idle state before Task Scheduler runs the task.

-IdleWaitTimeout <TimeSpan>

Specifies the amount of time that Task Scheduler waits for an idle condition to occur.

-MaintenanceDeadline <TimeSpan>

Specifies the amount of time after which Task Scheduler attempts to run the task during emergency Automatic maintenance, if the task failed to complete during regular Automatic maintenance.

-MaintenanceExclusive [<SwitchParameter>]

Indicates that the task needs to run alone when it is started in maintenance mode.

-MaintenancePeriod <TimeSpan>

Specifies the amount of time that the task needs to run once during regular Automatic maintenance.

-MultipleInstances <MultipleInstancesEnum>

Specifies the policy that defines how Task Scheduler handles multiple instances of the task. The acceptable values for this parameter are:

IgnoreNew. The new task instance is ignored. Parallel. The new task instance starts immediately. Queue. The new task instance starts as soon as the current instance completes.

-NetworkId <String>

Specifies the ID of a network profile that Task Scheduler uses to determine if the task can run. You must specify the ID of a network if you specify the RunOnlyIfNetworkAvailable parameter.

-NetworkName <String>

Specifies the name of a network profile that Task Scheduler uses to determine if the task can run. The Task Scheduler UI uses this setting for display purposes. Specify a network name if you specify the RunOnlyIfNetworkAvailable parameter.

-Priority <Int32>

Specifies the priority level of the task. Priority must be an integer from 0 (highest priority) to 10 (lowest priority). The default value is 7.

Priority levels 7 and 8 are used for background tasks. Priority levels 4, 5, and 6 are used for interactive tasks.

-RestartCount <Int32>

Specifies the number of times that Task Scheduler attempts to restart the task.

-RestartInterval <TimeSpan>

Specifies the amount of time that Task Scheduler attempts to restart the task.

-RestartOnIdle [<SwitchParameter>]

Indicates that Task Scheduler restarts the task when the computer cycles

into an idle condition more than once.

-RunOnlyIfIdle [<SwitchParameter>]

Indicates that Task Scheduler runs the task only when the computer is idle.

-RunOnlyIfNetworkAvailable [<SwitchParameter>]

Indicates that Task Scheduler runs the task only when a network is available. Task Scheduler uses the NetworkID parameter and NetworkName parameter that you specify in this cmdlet to determine if the network is available.

-StartWhenAvailable [<SwitchParameter>]

Indicates that Task Scheduler can start the task at any time after its scheduled time has passed.

-ThrottleLimit <Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then Windows PowerShell calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit applies only to the current cmdlet, not to the session or to the computer.

-WakeToRun [<SwitchParameter>]

Indicates that Task Scheduler wakes the computer before it runs the task.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [about_CommonParameters \(https://go.microsoft.com/fwlink/?LinkID=113216\)](https://go.microsoft.com/fwlink/?LinkID=113216).

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
PS C:\>$STSet = New-ScheduledTaskSettingsSet
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $STSet
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that use the default settings and assigns the ScheduledTaskSettings object to the \$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd and to use the default task settings.

This example registers a scheduled task that uses default task settings.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

----- Example 2: Set the priority of a scheduled task -----

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
PS C:\>$STSet = New-ScheduledTaskSettingsSet -Priority 5
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $Stset
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that sets a higher priority for the scheduled task, and assigns the ScheduledTaskSettings object to the

\$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd and to use the task settings that have a priority setting of 5.

This example sets the priority of a scheduled task.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

----- Example 3: Set restart settings for a scheduled task -----

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
PS C:\>$Stset = New-ScheduledTaskSettingsSet -RestartCount 3 -RestartInterval  
(New-TimeSpan -Minutes 60)
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $Stset
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that specify that Task Scheduler attempts three restarts of the task at sixty minute intervals. This command assigns the ScheduledTaskSettings object to the \$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd and to use the task settings that the ScheduledTaskSettings object defines.

This example sets restart settings for a scheduled task.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

----- Example 4: Set idle settings for a scheduled task -----

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
PS C:\>$Stset = New-ScheduledTaskSettingsSet -RunOnlyIfIdle -IdleDuration  
00:02:00 -IdleWaitTimeout 02:30:00
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $Stset
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that specify that Task Scheduler runs the task only when the computer is idle for 2 minutes and waits for 2 hours and 30 minutes for an idle condition. This command assigns the ScheduledTaskSettings object to the \$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd and to use the task settings that the ScheduledTaskSettings object defines.

This example sets idle settings for a scheduled task.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

Example 5: Register a scheduled task that runs only when a network is available

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
PS C:\>$Stset = New-ScheduledTaskSettingsSet -RunOnlyIfNetworkAvailable
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $Stset
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that specify that Task Scheduler runs the task only when a network is available. This command assigns the ScheduledTaskSettings object to the \$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd only when a network is available.

This example registers a scheduled task that runs only when a network is available.

Example 6: Register a scheduled task that has a time limit to complete the task

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"
```

```
$Stset = New-ScheduledTaskSettingsSet -ExecutionTimeLimit (New-TimeSpan -Hours  
1)
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Settings $Stset
```

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable.

The second command creates scheduled task settings that specify if the task is not finished after one hour, it is considered as failed. This command assigns the ScheduledTaskSettings object to the \$Stset variable.

The third command registers the scheduled task Task01 to run the task action named Cmd, only then finish the task after one hour.

Without the ExecutionTimeLimit setting defined, the time limit set to it's default of three days for the Task Scheduler is allowed to complete the task.

To configure the time limit, see `New-TimeSpan`
(`/powershell/module/microsoft.powershell.utility/new-timespan`).

REMARKS

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

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

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

For online help, type: `"get-help New-ScheduledTaskSettingsSet -online"`