



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-ScheduledTaskTrigger'

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

NAME

New-ScheduledTaskTrigger

SYNOPSIS

Creates a scheduled task trigger object.

SYNTAX

New-ScheduledTaskTrigger [-Once] [-AsJob] -At <DateTime> [-CimSession <CimSession[]>] [-RandomDelay <TimeSpan>] [-RepetitionDuration <TimeSpan>] [-RepetitionInterval <TimeSpan>] [-ThrottleLimit <Int32>] [<CommonParameters>]

New-ScheduledTaskTrigger [-Daily] [-AsJob] -At <DateTime> [-CimSession <CimSession[]>] [-DaysInterval <UInt32>] [-RandomDelay <TimeSpan>] [-ThrottleLimit <Int32>] [<CommonParameters>]

New-ScheduledTaskTrigger [-Weekly] [-AsJob] -At <DateTime> [-CimSession <CimSession[]>] [-DaysOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday}] [-RandomDelay <TimeSpan>] [-ThrottleLimit <Int32>] [-WeeksInterval <UInt32>] [<CommonParameters>]

```
New-ScheduledTaskTrigger [-AtLogOn] [-AsJob] [-CimSession <CimSession[]>]
[-RandomDelay <TimeSpan>] [-ThrottleLimit <Int32>] [-User <String>]
[<CommonParameters>]
```

```
New-ScheduledTaskTrigger [-AtStartup] [-AsJob] [-CimSession <CimSession[]>]
[-RandomDelay <TimeSpan>] [-ThrottleLimit <Int32>] [<CommonParameters>]
```

DESCRIPTION

The `New-ScheduledTaskTrigger` cmdlet creates and returns a new scheduled task trigger object.

You can use a time-based trigger or an event-based trigger to start a task.

Time-based triggers include starting a task at a specific time or starting a task multiple times on a daily or weekly schedule. Event-based triggers include starting a task when the system starts up or when a user logs on to the computer. Each task can contain one or more triggers, which means there are many ways that you can start a task. If a task has multiple triggers, Task Scheduler starts the task when any of the triggers occur.

PARAMETERS

`-AsJob` [<SwitchParameter>]

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

`-At` <DateTime>

Specifies a date and time to trigger the task. This parameter is valid for calendar-based triggers (Once, Daily, Weekly).

`-AtLogOn` [<SwitchParameter>]

Indicates that a trigger starts a task when a user logs on.

-AtStartup [<SwitchParameter>]

Indicates that a trigger starts a task when the system is started.

-CimSession <CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a Common Information Model (CIM) session object that represents a connection to a local computer or a remote computer, such as the output of a New-CimSession or Get-CimSession cmdlet. The default is the current session on the local computer.

-Daily [<SwitchParameter>]

Indicates that a trigger starts a task on a recurring daily schedule.

-DaysInterval <UInt32>

Specifies the interval between the days in the schedule. An interval of 1 produces a daily schedule. An interval of 2 produces an every-other day schedule.

-DaysOfWeek <DayOfWeek[]>

Specifies an array of the days of the week on which Task Scheduler runs the task.

-Once [<SwitchParameter>]

Indicates that a trigger starts a task once at a time specified in the At parameter.

-RandomDelay <TimeSpan>

Specifies a random amount of time to delay the start time of the trigger.

The delay time is a random time between the time the task triggers and the time that you specify in this setting.

-RepetitionDuration <TimeSpan>

Specifies how long the repetition pattern repeats after the task starts.

-RepetitionInterval <TimeSpan>

Specifies an amount of time between each restart of the task. The task will run, wait for the time interval specified, and then run again. This cycle continues for the time that you specify for the RepetitionDuration parameter.

-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.

-User <String>

Specifies the identifier of the user for a trigger that starts a task when a user logs on.

-Weekly [<SwitchParameter>]

Indicates that the trigger starts a task on a recurring weekly schedule.

-WeeksInterval <UInt32>

Specifies the interval between the weeks in the schedule. An interval of 1 produces a weekly schedule. An interval of 2 produces an every-other week schedule.

<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).

- Example 1: Register a scheduled task that starts a task once -

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"  
PS C:\>$Stt = New-ScheduledTaskTrigger -Once -At 3am  
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Trigger $Stt
```

This example registers a scheduled task that starts once.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable. The second command creates a scheduled task trigger that starts the task once at 3:00 A.M and assigns the ScheduledTaskTrigger object to the \$Stt variable. The third command registers the scheduled task Task01 to run the task action named Cmd once at 3:00 A.M.

-- Example 2: Register a scheduled task that starts every day --

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"  
PS C:\>$Stt = New-ScheduledTaskTrigger -Daily -At 3am  
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Trigger $Stt
```

This example registers a scheduled task that starts every day.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable. The second command creates a scheduled task trigger that starts every day at 3:00 A.M and assigns the ScheduledTaskTrigger object to the \$Stt variable. The third command registers the scheduled task Task01 to run the task action named Cmd every day at 3:00 A.M.

Example 3: Register a scheduled task that starts every 3 days

```
PS C:\>$Sta = New-ScheduledTaskAction -Execute "Cmd"  
PS C:\>$Stt = New-ScheduledTaskTrigger -Daily -DaysInterval 3 -At 3am  
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Trigger $Stt
```

This example registers a scheduled task that starts every 3 days.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable. The second command creates a scheduled task trigger that starts every 3 days at 3:00 A.M and assigns the ScheduledTaskTrigger object to the \$Stt variable. The third command registers the scheduled task Task01 to run the task action named cmd every 3 days at 3:00 A.M.

Example 4: Register a scheduled task that starts every-other week

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

```
PS C:\>$Stt = New-ScheduledTaskTrigger -Weekly -WeeksInterval 2 -DaysOfWeek  
Sunday -At 3am
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Trigger $Stt
```

This example registers a scheduled task that starts every other week.

The first command creates a scheduled task action named Cmd and assigns the ScheduledTaskAction object to the \$Sta variable. The second command creates a scheduled task trigger that starts every other Sunday at 3:00 A.M and assigns the ScheduledTaskTrigger object to the \$Stt variable. The third command registers the scheduled task Task01 to run the task action named Cmd every other Sunday at 3:00 A.M.

Example 5: Register a scheduled task that starts when a user logs on

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

```
PS C:\>$Stt = New-ScheduledTaskTrigger -AtLogon
```

```
PS C:\>Register-ScheduledTask Task01 -Action $Sta -Trigger $Stt
```

This example registers a scheduled task that starts when a user logs on.

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

scheduled task trigger that starts when a user logs on, and assigns the ScheduledTaskTrigger object to the \$Stt variable. The third command registers the scheduled task Task01 to run the task action named Cmd when a user logs on.

REMARKS

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

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

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

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