



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-JobTrigger'

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

NAME

New-JobTrigger

SYNOPSIS

Creates a job trigger for a scheduled job.

SYNTAX

New-JobTrigger [-Once] -At <System.DateTime> [-RandomDelay <System.TimeSpan>]
[-RepeatIndefinitely] [-RepetitionDuration <System.TimeSpan>]
[-RepetitionInterval <System.TimeSpan>] [<CommonParameters>]

New-JobTrigger [-Daily] -At <System.DateTime> [-DaysInterval <System.Int32>]
[-RandomDelay <System.TimeSpan>] [<CommonParameters>]

New-JobTrigger [-Weekly] -At <System.DateTime> -DaysOfWeek {Sunday | Monday |
Tuesday | Wednesday | Thursday | Friday | Saturday} [-RandomDelay
<System.TimeSpan>] [-WeeksInterval <System.Int32>] [<CommonParameters>]

New-JobTrigger [-AtLogOn] [-RandomDelay <System.TimeSpan>] [-User
<System.String>] [<CommonParameters>]

New-JobTrigger [-AtStartup] [-RandomDelay <System.TimeSpan>]
[<CommonParameters>]

DESCRIPTION

The `New-JobTrigger` cmdlet creates a job trigger that starts a scheduled job on a one-time or recurring schedule, or when an event occurs.`

You can use the `ScheduledJobTrigger` object that New-JobTrigger` returns to set a job trigger for a new or existing scheduled job. You can also create a job trigger with the Get-JobTrigger` cmdlet to get the job trigger of an existing scheduled job, or with a hash table value to represent a job trigger.`

When creating a job trigger, review the default values of the options specified by the `New-ScheduledJobOption` cmdlet. These options, which have the same valid and default values as the corresponding options in Task Scheduler , affect the scheduling and timing of scheduled jobs.`

`New-JobTrigger` 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

`-At <System.DateTime>`

Starts the job at the specified date and time. Enter a `DateTime` object,`

such as one that the ``Get-Date`` cmdlet returns, or a string that can be converted to a date and time, such as ``April 19, 2012 15:00``, ``12/31``, or ``3am``. If you don't specify an element of the date, such as the year, the date in the trigger has the corresponding element from the current date.

When using the `Once` parameter, set the value of the `At` parameter to a future date and time. Because the default date in a `DateTime` object is the current date, if you specify a time before the current time without an explicit date, the job trigger is created for a time in the past. `DateTime` objects, and strings that are converted to `DateTime` objects, are automatically adjusted to be compatible with the date and time formats selected for the local computer in `Region` and `Language` in Control Panel.

`-AtLogOn <System.Management.Automation.SwitchParameter>`

Starts the scheduled job when the specified users log on to the computer.

To specify a user, use the `User` parameter.

`-AtStartup <System.Management.Automation.SwitchParameter>`

Starts the scheduled job when Windows starts.

`-Daily <System.Management.Automation.SwitchParameter>`

Specifies a recurring daily job schedule. Use the other parameters in the `Daily` parameter set to specify the schedule details.

`-DaysInterval <System.Int32>`

Specifies the number of days between occurrences on a daily schedule. For example, a value of ``3`` starts the scheduled job on days ``1``, ``4``, ``7`` and so on. The default value is ``1``.

`-DaysOfWeek <System.DayOfWeek[]>`

Specifies the days of the week on which a weekly scheduled job runs. Enter day names, such as ``Monday`` or integers ``0``-``6``, where ``0`` represents Sunday. This parameter is required in the `Weekly` parameter set.

Day names are converted to their integer values in the job trigger. When you enclose day names in quotation marks in a command, enclose each day name in separate quotation marks, such as `"Monday", "Tuesday"`. If you enclose multiple day names in a single quotation mark pair, the corresponding integer values are summed. For example, `"Monday, Tuesday" (1 + 2)` results in a value of `Wednesday` (3).

-Once <System.Management.Automation.SwitchParameter>

Specifies a non-recurring (one time) or custom repeating schedule. To create a repeating schedule, use the Once parameter with the RepetitionDuration and RepetitionInterval parameters.

-RandomDelay <System.TimeSpan>

Enables a random delay that begins at the scheduled start time, and sets the maximum delay value. The length of the delay is set pseudo-randomly for each start and varies from no delay to the time specified by the value of this parameter. The default value, zero (00:00:00), disables the random delay.

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

-RepeatIndefinitely <System.Management.Automation.SwitchParameter>

This parameter, available starting in Windows PowerShell 4.0, eliminates the necessity of specifying a TimeSpan.MaxValue value for the RepetitionDuration parameter to run a scheduled job repeatedly, for an indefinite period.

-RepetitionDuration <System.TimeSpan>

Repeats the job until the specified time expires. The repetition frequency is determined by the value of the RepetitionInterval parameter. For

example, if the value of `RepetitionInterval` is 5 minutes and the value of `RepetitionDuration` is 2 hours, the job is triggered every five minutes for two hours.

Enter a timespan object, such as one that the `New-TimeSpan`` cmdlet returns or a string that can be converted to a timespan object, such as ``1:05:30``.

To run a job indefinitely, add the `RepeatIndefinitely` parameter instead.

To stop a job before the job trigger repetition duration expires, use the ``Set-JobTrigger`` cmdlet to set the `RepetitionDuration` value to zero (``0``).

This parameter is valid only when the `Once`, `At`, and `RepetitionInterval` parameters are used in the command.

`-RepetitionInterval <System.TimeSpan>`

Repeats the job at the specified time interval. For example, if the value of this parameter is 2 hours, the job is triggered every two hours. The default value, ``0``, does not repeat the job.

Enter a timespan object, such as one that the `New-TimeSpan`` cmdlet returns or a string that can be converted to a timespan object, such as ``1:05:30``.

This parameter is valid only when the `Once`, `At`, and `RepetitionDuration` parameters are used in the command.

`-User <System.String>`

Specifies the users who trigger an `AtLogon` start of a scheduled job. Enter the name of a user in ``<UserName>`` or ``<Domain\Username>`` format or enter an asterisk (``*``) to represent all users. The default value is all users.

-Weekly <System.Management.Automation.SwitchParameter>

Specifies a recurring weekly job schedule. Use the other parameters in the Weekly parameter set to specify the schedule details.

-WeeksInterval <System.Int32>

Specifies the number of weeks between occurrences on a weekly job schedule. For example, a value of `3` starts the scheduled job on weeks `1`, `4`, `7` and so on. The default value is `1`.

<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: Once Schedule -----

```
New-JobTrigger -Once -At "1/20/2012 3:00 AM"
```

The `New-JobTrigger` cmdlet to create a job trigger that starts a scheduled job only one time. The value of the At parameter is a string that Windows PowerShell converts into a DateTime object.

The At parameter value includes an explicit date, not just a time. If the date were omitted, the trigger would be created with the current date and 3:00 AM time, which is likely to represent a time in the past.

----- Example 2: Daily Schedule -----

```
New-JobTrigger -Daily -At "4:15 AM" -DaysInterval 3
```

This command creates a job trigger that starts a scheduled job every 3 days at 4:15 a.m.

Because the value of the At parameter does not include a date, the current date is used as the date value in the DateTime object. If the date and time is in the past, the scheduled job is started at the next occurrence, which is 3 days later from the At parameter value.

----- Example 3: Weekly Schedule -----

```
New-JobTrigger -Weekly -DaysOfWeek Monday, Wednesday, Friday -At "23:00"  
-WeeksInterval 4
```

Id	Frequency	Time	DaysOfWeek	Enabled
0	Weekly	9/21/2012 11:00:00 PM	{Monday, Wednesday, Friday}	True

This command creates a job trigger to start a scheduled job on Monday, Wednesday, and Friday at 2300 hours (11:00 PM) every 4 weeks.

You can also enter the DaysOfWeek parameter value in integers, such as ``-DaysOfWeek 1, 5``.

----- Example 4: Logon Schedule -----

```
New-JobTrigger -AtLogOn -User Domain01\Admin01
```

This command creates a job trigger to start a scheduled job whenever the domain administrator logs onto the computer.

----- Example 5: Using a Random Delay -----

```
New-JobTrigger -Daily -At 1:00 -RandomDelay 00:20:00
```

This command creates a job trigger to start a scheduled job every day at 1:00 in the morning. The command uses the RandomDelay parameter to set the maximum delay to 20 minutes. As a result, the job runs every day between 1:00 AM and 1:20 AM, with the interval varying pseudo-randomly.

You can use a random delay for sampling, load balancing, and other administrative tasks. When setting the delay value, review the effective and default values of the `New-ScheduledJobOption` cmdlet and coordinate the delay with the option settings.

--- Example 6: Create a Job Trigger for a New Scheduled Job ---

```
$t = New-JobTrigger -Weekly -DaysOfWeek 1,3,5 -At 12:01AM  
Register-ScheduledJob -Name Test-HelpFiles -FilePath  
C:\Scripts\Test-HelpFiles.ps1 -Trigger $t
```

The first command uses the `New-JobTrigger` cmdlet to create a job trigger that starts a job every Monday, Wednesday, and Friday at 12:01 AM. The command saves the job trigger in the `\$t` variable.

The second command uses the `Register-ScheduledJob` cmdlet to create a scheduled job that starts a job every Monday, Wednesday, and Friday at 12:01 AM. The value of the Trigger parameter is the trigger that is stored in the `\$t` variable.

----- Example 7: Add a Job Trigger to a Scheduled Job -----

```
Add-JobTrigger -Name SynchronizeApps -Trigger (New-JobTrigger -Daily -At  
3:10AM)
```

You can add multiple job triggers to any scheduled job.

The command uses the `Add-JobTrigger` cmdlet to add the job trigger to the SynchronizeApps scheduled job. The value of the Trigger parameter is a `New-JobTrigger` command that runs the job every day at 3:10 AM.

When the command completes, SynchronizeApps is a scheduled job that runs at the times specified by the job trigger.

----- Example 8: Create a repeating job trigger -----


```
New-JobTrigger -Once -At "09/12/2013 1:00:00" -RepetitionInterval  
(New-TimeSpan -Hours 1) -RepetitionDuration (New-Timespan -Hours 48)
```

This command creates a job trigger that runs a job every 60 minutes for 48 hours beginning on September 12, 2013 at 1:00 AM.

----- Example 9: Stop a repeating job trigger -----

```
Get-JobTrigger -Name SecurityCheck | Set-JobTrigger -RepetitionInterval 0:00  
-RepetitionDuration 0:00
```

This command forcibly stops the SecurityCheck job, which is triggered to run every 60 minutes until its job trigger expires.

To prevent the job from repeating, the command uses the `Get-JobTrigger`` to get the job trigger of the SecurityCheck job and the `Set-JobTrigger`` cmdlet to change the repetition interval and repetition duration of the job trigger to zero (`0``).

----- Example 10: Create an hourly job trigger -----

```
New-JobTrigger -Once -At "9/21/2012 0am" -RepetitionInterval (New-TimeSpan  
-Hour 12) -RepetitionDuration ([TimeSpan]::MaxValue)
```

The following command creates a job trigger that runs a scheduled job once every 12 hours for an indefinite period of time. The schedule begins tomorrow (9/21/2012) at midnight (0:00 AM).

REMARKS

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

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

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

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