PowerShell Get-Help on command 'Unregister-ScheduledJob'

PS C:\Users\wahid> Get-Help Unregister-ScheduledJob

```
NAME
    Unregister-ScheduledJob

SYNOPSIS
    Deletes scheduled jobs on the local computer.


SYNTAX
    Unregister-ScheduledJob [-Id] <System.Int32[]> [-Force] [-Confirm] [-WhatIf]
    [<CommonParameters>]

    Unregister-ScheduledJob [-InputObject]
    <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition[]> [-Force]
    [-Confirm] [-WhatIf] [<CommonParameters>]

    Unregister-ScheduledJob [-Name] <System.String[]> [-Force] [-Confirm]
    [-WhatIf] [<CommonParameters>]


DESCRIPTION
    The `Unregister-ScheduledJob` cmdlet deletes scheduled jobs from the local
```

computer.

When it deletes or unregisters a scheduled job, `Unregister-ScheduledJob` deletes the directory for the scheduled job (in the `$HOME\AppData\Local\Microsoft\Windows\PowerShell\ScheduledJobs directory`), which contains the XML file that defines the scheduled job, the job execution history, and all job results. This action also deletes the job from Task Scheduler.

`Unregister-ScheduledJob` deletes only the scheduled jobs that are created by using the `Register-ScheduledJob` cmdlet. It does not delete scheduled jobs that are created in Task Scheduler.

You can use the parameters of `Unregister-ScheduledJob` to delete scheduled jobs by ID or name, or pipe scheduled jobs from `Get-ScheduledJob` to `Unregister-ScheduledJob`.

`Unregister-ScheduledJob` 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

-Force <System.Management.Automation.SwitchParameter>

Deletes the scheduled job even if an instance of the job is running. By default, `Unregister-ScheduledJob` does not interrupt running jobs.

-Id <System.Int32[]>

Deletes the scheduled jobs with the specified identification numbers (ID). Enter the IDs of scheduled jobs on the computer.

-InputObject <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition[]>

Specifies a scheduled job. Enter a variable that contains ScheduledJob objects or type a command or expression that gets ScheduledJob objects, such as a `Get-ScheduledJob` command. You can also pipe ScheduledJob objects to `Unregister-JobTrigger`.

-Name <System.String[]>

Deletes the scheduled jobs with the specified names. Enter the names of one or more scheduled jobs on the computer. Wildcards are supported.

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<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: Delete a scheduled job --------------

```
Unregister-ScheduledJob TestJob
```

This command deletes the TestJob scheduled job on the local computer.

------------- Example 2: Delete all scheduled jobs -------------

```
Get-ScheduledJob | Unregister-ScheduledJob -Force
Unregister-ScheduledJob -Name "*" -Force
```

This example shows two different commands that delete all scheduled jobs on the local computer.

The first command uses the `Get-ScheduledJob` cmdlet to get all scheduled jobs on the local computer. A pipeline operator (`|`) sends the scheduled jobs to `Unregister-ScheduleJob`, which deletes them.

The second command uses the Name parameter of `Unregister-ScheduledJob` with a value of all (`*`) to delete all scheduled jobs.

Both commands use the Force parameter, which deletes a scheduled job even if an instance of the job is running.

---- Example 3: Delete a scheduled job on a remote computer ----

```
Invoke-Command -ComputerName "Server01" { Unregister-ScheduledJob -Name
"Test*"}
```

This command deletes scheduled jobs with names that begin with Test on the Server01 remote computer. The command uses the `Invoke-Command` cmdlet to run the `Unregister-ScheduledJob` command on the Server02 computer.

REMARKS

To see the examples, type: "get-help Unregister-ScheduledJob -examples".

For more information, type: "get-help Unregister-ScheduledJob -detailed".

For technical information, type: "get-help Unregister-ScheduledJob -full".

For online help, type: "get-help Unregister-ScheduledJob -online"