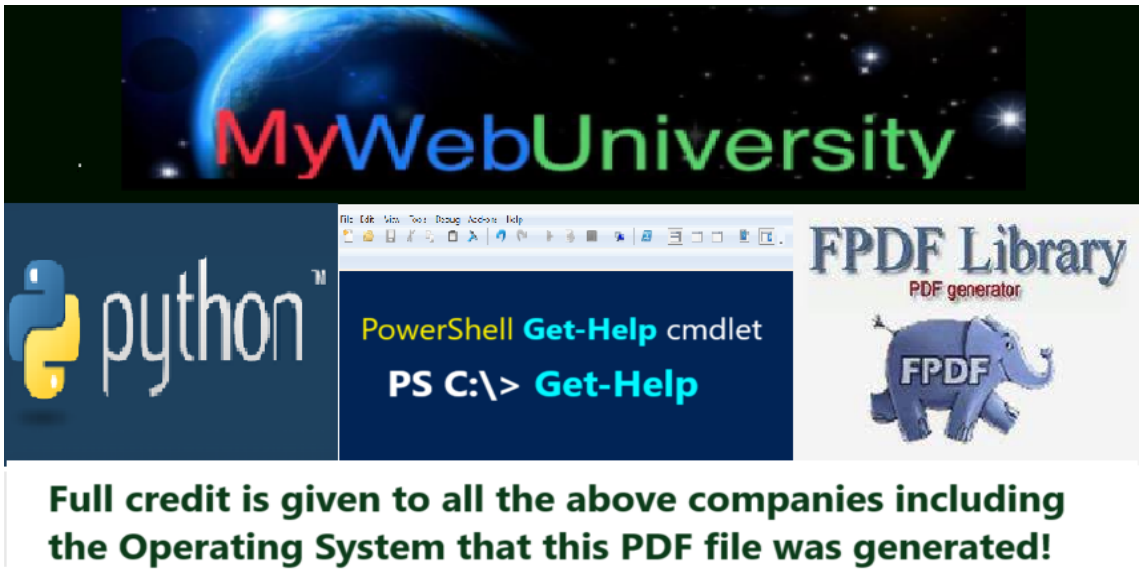

Windows PowerShell Get-Help on Cmdlet 'Get-JobTrigger'

PS:\>Get-HELP Get-JobTrigger -Full

NAME

Get-JobTrigger

SYNOPSIS

Gets the job triggers of scheduled jobs.

SYNTAX

Get-JobTrigger [-Id] <System.Int32> [[-TriggerId] <System.Int32[]>] [<CommonParameters>]

Get-JobTrigger [-InputObject] <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [[-TriggerId] <System.Int32[]>] [<CommonParameters>]

Get-JobTrigger [-Name] <System.String> [[-TriggerId] <System.Int32[]>] [<CommonParameters>]

DESCRIPTION

The `Get-JobTrigger` cmdlet gets the job triggers of scheduled jobs. You can use this command to examine the job triggers or to pipe the job triggers to other cmdlets.

A job trigger defines a recurring schedule or conditions for starting a scheduled job. Job triggers are not saved to disk independently; they are part of a scheduled

job. To get a job trigger, specify the scheduled job that the trigger starts.

Use the parameters of the `Get-JobTrigger` cmdlet to identify the scheduled jobs. You can identify the scheduled jobs by their names or identification numbers, or by

entering or piping ScheduledJob objects, such as those that are returned by the `Get-ScheduledJob` cmdlet, to `Get-JobTrigger`.

`Get-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

-Id <System.Int32>

Specifies the identification number of a scheduled job. `Get-JobTrigger` gets the job trigger of the specified scheduled job.

To get the identification number of scheduled jobs on the local computer or a remote computer, use the `Get-ScheduledJob` cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-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 `Get-JobTrigger`.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Name <System.String>

Specifies the name of a scheduled job. `Get-JobTrigger` gets the job trigger of the specified scheduled job. Wildcards are supported.

To get the names of scheduled jobs on the local computer or a remote computer, use the `Get-ScheduledJob` cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-TriggerId <System.Int32[]>

Gets the specified job triggers. Enter the trigger IDs of one or more job triggers of a scheduled job. Use this parameter when the scheduled job that is specified

by the Name , ID , or InputObject parameters has multiple job triggers.

Required? false

Position? 1

Default value None

Accept pipeline input? False

Accept wildcard characters? false

<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).

INPUTS

Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition

You can pipe a scheduled job to this cmdlet.

OUTPUTS

Microsoft.PowerShell.ScheduledJob.ScheduledJobTrigger

This cmdlet returns the scheduled job's trigger.

NOTES

------ Example 1: Get a job trigger by scheduled job name ------

Get-JobTrigger -Name "BackupJob"

The command uses the Name parameter of `Get-JobTrigger` to get the job triggers of the `BackupJob` scheduled job.

---------- Example 2: Get a job trigger by ID ----------

Get-ScheduledJob

Id	Name	Triggers	Command	Enabled
--	----	--------	-------	-------

1 ArchiveProjects {1} \\Server\Share\Archive-Projects.ps1 True

2 Backup {1,2} \\Server\Share\Run-Backup.ps1 True

3 Test-HelpFiles {1} \\Server\Share\Test-HelpFiles.ps1 True

4 TestJob {} \\Server\Share\Run-AllTests.ps1 True

Get-JobTrigger -ID 3

The first command uses the `Get-ScheduledJob` cmdlet to display the scheduled jobs on the local computer. The display includes the IDs of the scheduled jobs.

The second command uses the `Get-JobTrigger` cmdlet to get the job trigger for the `Test-HelpFiles` job (whose ID is `3`).

--------- Example 3: Get job triggers by piping a job ---------

Get-ScheduledJob -Name *Backup*, *Archive* | Get-JobTrigger

This command gets the job triggers of all jobs that have `Backup` or `Archive` in their names.

- Example 4: Get the job trigger of a job on a remote computer -

Invoke-Command -ComputerName Server01 { Get-ScheduledJob Backup | Get-JobTrigger -TriggerID 2 }

The command uses the `Invoke-Command` cmdlet to run a command on the Server01 computer. It uses the `Get-ScheduledJob` cmdlet to get the `Backup` scheduled job, which

it pipes to the `Get-JobTrigger` cmdlet. It uses the TriggerID parameter to get only the second trigger.

---------- Example 5: Get all job triggers ----------

Get-ScheduledJob | Get-JobTrigger |

Format-Table -Property ID, Frequency, At, DaysOfWeek, Enabled, @{Label="ScheduledJob";Expression={$_.JobDefinition.Name}} -AutoSize

Id Frequency At DaysOfWeek Enabled ScheduledJob

-- --------- -- ---------- ------- ------------

1 Weekly 9/28/2011 3:00:00 AM {Monday} True Backup

1 Daily 9/27/2011 11:00:00 PM True Test-HelpFiles

The command uses the `Get-ScheduledJob` to get the scheduled jobs on the local computer and pipes them to `Get-JobTrigger`, which gets the job trigger of each

scheduled job (if any).

To add the name of the scheduled job to the job trigger display, the command uses the calculated property feature of the `Format-Table` cmdlet. In addition to the job

trigger properties that are displayed by default, the command creates a new ScheduledJob property that displays the name of the scheduled job.

-- Example 6: Get the job trigger property of a scheduled job --

```
(Get-ScheduledJob Test-HelpFiles).JobTriggers
Get-ScheduledJob | foreach {$_.JobTriggers}
```

The first command uses the `Get-ScheduledJob` cmdlet to get the `Test-HelpFiles` scheduled job. Then it uses the dot method (`.`) to get the JobTriggers property of

the `Test-HelpFiles` scheduled job.

The second command uses the `Get-ScheduledJob` cmdlet to get all scheduled jobs on the local computer. It uses the `ForEach-Object` cmdlet to get the value of the

JobTriggers property of each scheduled job.

The job triggers of a scheduled job are stored in the JobTriggers property of the job. This example shows alternatives to using the `Get-JobTrigger` cmdlet to get job

triggers. The results are identical to using the `Get-JobTrigger` cmdlet and the techniques can be used interchangeably.

---------- Example 7: Compare job triggers ----------

```
Get-ScheduledJob -Name ArchiveProjects | Get-JobTrigger | Tee-Object -Variable t1

Id       Frequency      Time                DaysOfWeek          Enabled
--       ---------      ----                ----------          -------
0        Daily          9/26/2011 3:00:00 AM                    True
```

```
Get-ScheduledJob -Name "Test-HelpFiles" | Get-JobTrigger | Tee-Object -Variable t2

Id        Frequency       Time                 DaysOfWeek            Enabled
--        ---------       ----                 ----------            -------
0         Daily           9/26/2011 3:00:00 AM                       True

$t1| Get-Member -Type Property | ForEach-Object { Compare-Object $t1 $t2 -Property $_.Name}

RandomDelay                                SideIndicator
-----------                               -------------
00:00:00                                     =>
00:03:00                                     <=
```

The first command gets the job trigger of the `ArchiveProjects` scheduled job. The command pipes the job trigger to the `Tee-Object` cmdlet, which saves the job trigger in the `$t1` variable and displays it at the command line.

The second command gets the job trigger of the `Test-HelpFiles` scheduled job. The command pipes the job trigger to the `Tee-Object` cmdlet, which saves the job trigger in the `$t2` variable and displays it at the command line.

The third command compares the job triggers in the `$t1` and $t2 variables. It uses the `Get-Member` cmdlet to get the properties of the job trigger in the $t1 variable. It pipes the properties to the `ForEach-Object` cmdlet, which compares each property to the properties of the job trigger in the `$t2` variable by name. The command then pipes the differing properties to the `Format-List` cmdlet, which displays them in a list.The output indicates that, although the job triggers appear to be the same, the `HelpFiles` job trigger includes a random delay of three (`3`) minutes.

This example shows how to compare the job triggers of two scheduled jobs.

https://learn.microsoft.com/powershell/module/psscheduledjob/get-jobtrigger?view=powershell-5.1&WT.mc_id=ps-gethelp

Add-JobTrigger

Disable-JobTrigger

Disable-ScheduledJob

Enable-JobTrigger

Enable-ScheduledJob

Get-JobTrigger

Get-ScheduledJob

Get-ScheduledJobOption

New-JobTrigger

New-ScheduledJobOption

Register-ScheduledJob

Remove-JobTrigger

Set-JobTrigger

Set-ScheduledJob

Set-ScheduledJobOption

Unregister-ScheduledJob