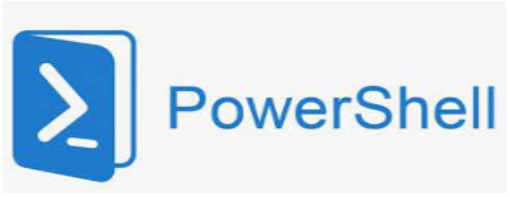

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Enable-ScheduledJob'

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

NAME

Enable-ScheduledJob

SYNOPSIS

Enables a scheduled job.

SYNTAX

Enable-ScheduledJob [-Id] <System.Int32> [-PassThru] [-Confirm] [-WhatIf]
[<CommonParameters>]

Enable-ScheduledJob [-InputObject]
<Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition> [-PassThru]
[-Confirm] [-WhatIf] [<CommonParameters>]

Enable-ScheduledJob [-Name] <System.String> [-PassThru] [-Confirm] [-WhatIf]
[<CommonParameters>]

DESCRIPTION

The `Enable-ScheduledJob` cmdlet re-enables scheduled jobs that are disabled,

such as those that are disabled by using the `Disable-ScheduledJob` cmdlet. Enabled jobs run automatically when triggered.

To enable a scheduled job, the `Enable-ScheduledJob` cmdlet sets the Enabled property of the scheduled job to `$true`.

`Enabled-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

-Id <System.Int32>

Enables the scheduled job with the specified identification number (ID). Enter the ID of a scheduled job.

-InputObject <Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition>

Specifies the scheduled job to enable. Enter a variable that contains ScheduledJobDefinition objects or type a command or expression that gets ScheduledJobDefinition objects, such as a `Get-ScheduledJob` command. You can also pipe a ScheduledJobDefinition object to `Enable-ScheduledJob`.

-Name <System.String>

Enables the scheduled jobs with the specified names. Enter the name of a scheduled job. Wildcards are supported.

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.

-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: Enable a scheduled job ----------

Enable-ScheduledJob -ID 2 -Passthru

The `Enable-ScheduledJob` command enables the scheduled job with ID 2 on the local computer. The PassThru parameter allows the Job object to be output.

---------- Example 2: Enable all scheduled jobs ----------

Get-ScheduledJob | Enable-ScheduledJob

The `Get-ScheduledJob` cmdlet gets all scheduled jobs and pipes them to `Enable-ScheduledJob` cmdlet to enable them.

`Enable-ScheduledJob` does not generate warnings or errors if you enable a scheduled job that is already enabled, so you can enable all scheduled jobs without conditions.

---------- Example 3: Enable selected scheduled jobs ----------

```
Get-ScheduledJob | Get-ScheduledJobOption | Where-Object
{$_.RunWithoutNetwork} |
    ForEach-Object {Enable-ScheduledJob -InputObject $_.JobDefinition}
```

The command uses the `Get-ScheduledJob` cmdlet to get all scheduled jobs on the computer. A pipeline operator (`|`) sends the scheduled jobs to the `Get-ScheduledJobOption` cmdlet, which gets the job options of each scheduled job. Each job options object has a JobDefinition property that contains the associated scheduled job. The JobDefinition property is used to complete the command.

The command uses a pipeline operator (`|`) to send the job options to the `Where-Object` cmdlet, which selects scheduled job option objects in which the RunWithoutNetwork property has a value of `$true`. Another pipeline operator sends the selected scheduled job options objects to the `ForEach-Object` cmdlet which runs an `Enable-ScheduledJob` command on the scheduled job in the value of the JobDefinition property of each job options object.

---- Example 4: Enable scheduled jobs on a remote computer ----

```
Invoke-Command -ComputerName "Srv01,Srv10" -ScriptBlock {Enable-ScheduledJob
-Name "Inventory"}
```

This command enables scheduled jobs that have "test" in their names on two remote computers, Srv01 and Srv10.

The command uses the `Invoke-Command` cmdlet to run an `Enable-ScheduledJob` command on the Srv01 and Srv10 computers. The command uses the Name parameter of `Enable-ScheduledJob` to enable the Inventory scheduled job on each computer.

REMARKS

To see the examples, type: "get-help Enable-ScheduledJob -examples".
For more information, type: "get-help Enable-ScheduledJob -detailed".
For technical information, type: "get-help Enable-ScheduledJob -full".

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