PowerShell Get-Help on command 'Resume-Service'

PS C:\Users\wahid> Get-Help Resume-Service

NAME

    Resume-Service

SYNOPSIS

    Resumes one or more suspended (paused) services.

SYNTAX

    Resume-Service -DisplayName <System.String[]> [-Exclude <System.String[]>]

    [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf]

    [<CommonParameters>]

    Resume-Service [-InputObject] <System.ServiceProcess.ServiceController[]>

    [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru]

    [-Confirm] [-WhatIf] [<CommonParameters>]

    Resume-Service [-Name] <System.String[]> [-Exclude <System.String[]>]

    [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf]

    [<CommonParameters>]

DESCRIPTION

The `Resume-Service` cmdlet sends a resume message to the Windows Service
Controller for each of the specified services. If a service is suspended, it
resumes. If it is currently running, the message is ignored. You can specify
the services by their service names or display names, or you can use the
InputObject parameter to pass a service object that represents the services
that you want to resume.

PARAMETERS

-DisplayName <System.String[]>

Specifies the display names of the services to be resumed. Wildcard
characters are permitted.

-Exclude <System.String[]>

Specifies services that this cmdlet omits. The value of this parameter
qualifies the Name parameter. Enter a name element or pattern, such as s*.
Wildcard characters are permitted.

-Include <System.String[]>

Specifies services to resume. The value of this parameter qualifies Name
parameter. Enter a name element or pattern, such as s*. Wildcard
characters are permitted.

-InputObject <System.ServiceProcess.ServiceController[]>

Specifies ServiceController objects that represent the services to
resumed. Enter a variable that contains the objects, or type a command or
expression that gets the objects.

-Name <System.String[]>

Specifies the service names of the services to be resumed.

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object that represents the service. 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: Resume a service on the local computer ------

PS C:\> Resume-Service "sens"

This command resumes the System Event Notification service on the local

computer. The service name is represented in the command by sens. The command

uses the Name parameter to specify the service name of the service, but the

command omits the parameter name because the parameter name is optional.

----------- Example 2: Resume all suspended services -----------

PS C:\> Get-Service | Where-Object {$_.Status -eq "Paused"} | Resume-Service

This command resumes all of the suspended services on the computer. The

`Get-Service` cmdlet command gets all of the services on the computer. The

pipeline operator (`|`) passes the results to the `Where-Object` cmdlet, which

selects the services that have a Status property of Paused. The next pipeline

operator sends the results to `Resume-Service`, which resumes the paused

services.

In practice, you would use the WhatIf parameter to determine the effect of the command before you run it.

REMARKS

To see the examples, type: "get-help Resume-Service -examples".

For more information, type: "get-help Resume-Service -detailed".

For technical information, type: "get-help Resume-Service -full".

For online help, type: "get-help Resume-Service -online"