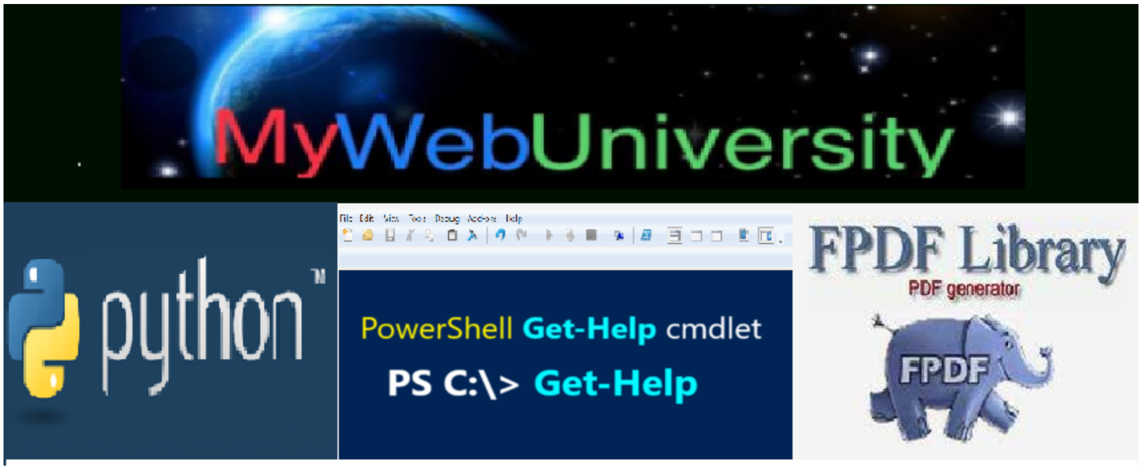

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Suspend-Service'

PS:\>Get-HELP Suspend-Service -Full

NAME

Suspend-Service

SYNOPSIS

Suspends (pauses) one or more running services.

SYNTAX

Suspend-Service -DisplayName <System.String[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Suspend-Service [-InputObject] <System.ServiceProcess.ServiceController[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf]

[<CommonParameters>]

Suspend-Service [-Name] <System.String[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Suspend-Service` cmdlet sends a suspend message to the Windows Service Controller for each of the specified services. While suspended, the service is still running, but its action is stopped until resumed, such as by usingthe `Resume-Service` cmdlet. 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 suspend.

PARAMETERS

-DisplayName <System.String[]>

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

Required?   true

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-Exclude <System.String[]>

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

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-Include <System.String[]>

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

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-InputObject <System.ServiceProcess.ServiceController[]>

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

the objects.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByValue)

Accept wildcard characters? false

-Name <System.String[]>

Specifies the service names of the services to suspend. Wildcard characters are permitted.

The parameter name is optional. You can use Name or its alias, ServiceName , or you can omit the parameter name.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByPropertyName, ByValue)

Accept wildcard characters? true

-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.

Required?   false

```
        Position?                    named
        Default value                False
        Accept pipeline input?       False
        Accept wildcard characters?  false

    -Confirm <System.Management.Automation.SwitchParameter>
        Prompts you for confirmation before running the cmdlet.

        Required?                    false
        Position?                    named
        Default value                False
        Accept pipeline input?       False
        Accept wildcard characters?  false

    -WhatIf <System.Management.Automation.SwitchParameter>
        Shows what would happen if the cmdlet runs. The cmdlet is not run.

        Required?                    false
        Position?                    named
        Default value                False
        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
    System.ServiceProcess.ServiceController
        You can pipe a service object to this cmdlet.
```

System.String

You can pipe a string that contains a service name to this cmdlet.

OUTPUTS

None

By default, this cmdlet returns no output.

System.ServiceProcess.ServiceController

When you use the PassThru parameter, this cmdlet returns a ServiceController object representing the service.

NOTES

- `Suspend-Service` can control services only when the current user has permission to do this. If a command does not work correctly, you might not have the

required permissions. - `Suspend-Service` can suspend only services that support being suspended and resumed. To determine whether a particular service can be

suspended, use the `Get-Service` cmdlet together with the CanPauseAndContinue property. For example, `Get-Service wmi | Format-List Name, CanPauseAndContinue`.

To find all services on the computer that can be suspended, type `Get-Service | Where-Object {$_.CanPauseAndContinue -eq $true}`. - To find the service names

and display names of the services on your system, type `Get-Service`. The service names appear in the Name column, and the display names appear in the

DisplayName column.

----------------- Example 1: Suspend a service -----------------

```
PS C:\> Suspend-Service -DisplayName "Telnet"
```

This command suspends the Telnet service (Tlntsvr) service on the local computer.

- Example 2: Display what would happen if you suspend services -

```
PS C:\> Suspend-Service -Name lanman* -WhatIf
```

This command tells what would happen if you suspended the services that have a service name that starts with lanman. To suspend the services, rerun the command

without the WhatIf parameter.

------------- Example 3: Get and suspend a service -------------

```
PS C:\> Get-Service schedule | Suspend-Service
```

This command uses the `Get-Service` cmdlet to get an object that represents the Task Scheduler (Schedule) service on the computer. The pipeline operator (`|`) passes

the result to `Suspend-Service`, which suspends the service.

---- Example 4: Suspend all services that can be suspended ----

```
PS C:\> Get-Service | Where-Object {$_.CanPauseAndContinue -eq "True"} | Suspend-Service -Confirm
```

This command suspends all of the services on the computer that can be suspended. It uses `Get-Service` to get objects that represent the services on the computer. The

pipeline operator passes the results to the `Where-Object` cmdlet, which selects only the services that have a value of `$True` for the CanPauseAndContinue property.

Another pipeline operator passes the results to `Suspend-Service`. The Confirm parameter prompts you for confirmation before suspending each of the services.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/suspend-service?view=powershell-5.1&WT.mc_id=ps-gethelp

Get-Service

New-Service

Restart-Service

Resume-Service

Set-Service

Start-Service

Stop-Service