

Version

- 6
- 5.1
- 5.0
- 4.0
- 3.0

Search

Stop-Service

Module: [Microsoft.PowerShell.Management](#)

Stops one or more running services.

In this article

- [Syntax](#)
- [Description](#)
- [Examples](#)
- [Required Parameters](#)
- [Optional Parameters](#)
- [Inputs](#)
- [Outputs](#)
- [Notes](#)
- [Related Links](#)

PowerShell

Copy

```
Stop-Service
    [-Force]
    [-NoWait]
    [-InputObject] <ServiceController[]>
    [-PassThru]
    [-Include <String[]>]
    [-Exclude <String[]>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

PowerShell

Copy

```
Stop-Service
    [-Force]
    [-NoWait]
    [-Name] <String[]>
    [-PassThru]
    [-Include <String[]>]
```

```
[-Exclude <String[]>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

PowerShell

 Copy

```
Stop-Service
[-Force]
[-NoWait]
[-PassThru]
-DisplayName <String[]>
[-Include <String[]>]
[-Exclude <String[]>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

Description

The **Stop-Service** cmdlet sends a stop message to the Windows Service Controller for each of the specified services. 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 service that you want to stop.

Examples

Example 1: Stop a service on the local computer

PowerShell

 Copy

```
PS C:\> Stop-Service -Name "sysmonlog"
```

This command stops the Performance Logs and Alerts (SysmonLog) service on the local computer.

Example 2: Stop a service by using the display name

PowerShell

 Copy

```
PS C:\> Get-Service -DisplayName "telnet" | Stop-Service
```

This command stops the Telnet service on the local computer. The command uses **Get-Service** to get an object that represents the Telnet service. The pipeline operator (|) pipes the object to **Stop-Service**, which stops the service.

Example 3: Stop a service that has dependent services

```
PS C:\> Get-Service -Name "iisadmin" | Format-List -Property Name, DependentServices
PS C:\> Stop-Service -Name "iisadmin" -Force -Confirm
```

This example stops the IISAdmin service on the local computer. Because stopping this service also stops the services that depend on the IISAdmin service, it is best to precede **Stop-Service** with a command that lists the services that depend on the IISAdmin service.

The first command lists the services that depend on IISAdmin. It uses **Get-Service** to get an object that represents the IISAdmin service. The pipeline operator (|) passes the result to the Format-List cmdlet. The command uses the *Property* parameter of **Format-List** to list only the **Name** and **DependentServices** properties of the service.

The second command stops the IISAdmin service. The *Force* parameter is required to stop a service that has dependent services. The command uses the *Confirm* parameter to request confirmation from the user before it stops each service.

Required Parameters

-DisplayName

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

Type:
String[]

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-InputObject

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

Type:
ServiceController[]

Position:
0

Default value:
None

Accept pipeline input:
True (ByValue)

Accept wildcard characters:
False

-Name

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

Type:
String[]

Aliases:
ServiceName

Position:
0

Default value:
None

Accept pipeline input:
True (ByPropertyName, ByValue)

Accept wildcard characters:
False

Optional Parameters

-Confirm

Prompts you for confirmation before running the cmdlet.

Type:
SwitchParameter

Aliases:
cf

Position:
Named

Default value:
False

Accept pipeline input:
False

Accept wildcard characters:
False

-Exclude

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.

Type:
String[]

Position:
Named

Default value:
None

Accept pipeline input:
False

Accept wildcard characters:
False

-Force

Forces the cmdlet to stop a service even if that service has dependent services.

Type:
SwitchParameter

Position:

Named

Default value:

None

Accept pipeline input:

False

Accept wildcard characters:

False

-Include

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

Type:

String[]

Position:

Named

Default value:

None

Accept pipeline input:

False

Accept wildcard characters:

False

-NoWait

Indicates that this cmdlet uses the no wait option.

Type:

SwitchParameter

Position:

Named

Default value:

None

Accept pipeline input:

False

Accept wildcard characters:

False

-PassThru

Returns an object that represents the service. By default, this cmdlet does not generate any output.

Type:

SwitchParameter

Position:

Named

Default value:

None

Accept pipeline input:

False

Accept wildcard characters:

False

-WhatIf

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

Type:

SwitchParameter

Aliases:

wi

Position:

Named

Default value:

False

Accept pipeline input:

False

Accept wildcard characters:

False

Inputs

System.ServiceProcess.ServiceController, **System.String**

You can pipe a service object or a string that contains the name of a service to this cmdlet.

Outputs

None, **System.ServiceProcess.ServiceController**

This cmdlet generates a **System.ServiceProcess.ServiceController** object that represents the service, if you use the *PassThru* parameter. Otherwise, this cmdlet does not generate any output.

Notes

- You can also refer to **Stop-Service** by its built-in alias, **spsv**. For more information, see [about_Aliases](#).

Stop-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.

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.

-

Related Links

- [Get-Service](#)
- [New-Service](#)
- [Restart-Service](#)
- [Resume-Service](#)
- [Set-Service](#)
- [Start-Service](#)
- [Suspend-Service](#)