Set-Service

Reference

Feedback

Module: Microsoft.PowerShell.Management

In this article

Starts, stops, and suspends a service, and changes its properties.

Syntax

```
Set-Service
   [-Name] <String>
   [-DisplayName <String>]
   [-Credential <PSCredential>]
   [-Description <String>]
   [-StartupType <ServiceStartupType>]
   [-Status <String>]
   [-SecurityDescriptorSddl <String>]
   [-Force]
   [-PassThru]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

```
Set-Service
   [-InputObject] <ServiceController>
   [-DisplayName <String>]
   [-Credential <PSCredential>]
   [-Description <String>]
   [-StartupType <ServiceStartupType>]
   [-SecurityDescriptorSddl <String>]
   [-Status <String>]
   [-Force]
   [-PassThru]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Description

> **This cmdlet is only available on the Windows platform.**

The `Set-Service` cmdlet changes the properties of a service such as the **Status**, **Description**, **DisplayName**, and **StartupType**. `Set-Service` can start, stop, suspend, or pause a service. To identify a service, enter its service name or submit a service object. Or, send a service name or service object down the pipeline to `Set-Service`.

Examples

Example 1: Change a display name

In this example, a service's display name is changed. To view the original display name, use `Get-Service`.

```
Set-Service -Name LanmanWorkstation -DisplayName "LanMan Wor
```

`Set-Service` uses the **Name** parameter to specify the service's name, **LanmanWorkstation**. The **DisplayName** parameter specifies the new display name, **LanMan Workstation**.

Example 2: Change the startup type of services

This example shows how to change a service's startup type.

```
Set-Service -Name BITS -StartupType Automatic
Get-Service BITS | Select-Object -Property Name, StartType,

Name  StartType   Status
----  ---------   ------
BITS  Automatic  Running
```

`Set-Service` uses the **Name** parameter to specify the service's name, **BITS**. The **StartupType** parameter sets the service to **Automatic**.

`Get-Service` uses the **Name** parameter to specify the **BITS** service and sends the object down the pipeline. `Select-Object` uses the **Property** parameter to display the **BITS** service's status.

Example 3: Change the description of a service

This example changes the BITS service's description and displays the result.

The `Get-CimInstance` cmdlet is used because it returns a **Win32_Service** object that includes the service's **Description**.

```
Get-CimInstance Win32_Service -Filter 'Name = "BITS"'  | For

Name        : BITS
Description : Transfers files in the background using idle r
              disabled, then any applications that depend or
              Explorer, will be unable to automatically dowr

Set-Service -Name BITS -Description "Transfers files in the 
Get-CimInstance Win32_Service -Filter 'Name = "BITS"' | Forn

Name        : BITS
Description : Transfers files in the background using idle r
```

`Get-CimInstance` sends the object down the pipeline to `Format-List` and displays the service's name and description. For comparison purposes, the command is run before and after the description is updated.

`Set-Service` uses the **Name** parameter to specify the **BITS** service. The **Description** parameter specifies the updated text for the services' description.

Example 4: Start a service

In this example, a service is started.

```
Set-Service -Name WinRM -Status Running -PassThru

Status   Name               DisplayName
------   ----               -----------
Running  WinRM              Windows Remote Management (WS-Ma
```

`Set-Service` uses the **Name** parameter to specify the service, **WinRM**. The **Status** parameter uses the value **Running** to start the service. The **PassThru** parameter outputs a **ServiceController** object that displays the results.

Example 5: Suspend a service

This example uses the pipeline to pause to service.

```
Get-Service -Name Schedule | Set-Service -Status Paused
```

`Get-Service` uses the **Name** parameter to specify the **Schedule** service, and sends the object down the pipeline. `Set-Service` uses the **Status** parameter to set the service to **Paused**.

Example 6: Stop a service

This example uses a variable to stop a service.

```
$S = Get-Service -Name Schedule
Set-Service -InputObject $S -Status Stopped
```

`Get-Service` uses the **Name** parameter to specify the service, **Schedule**. The object is stored in the variable, `$S`. `Set-Service` uses the **InputObject** parameter and specifies the object stored `$S`. The **Status** parameter sets the service to **Stopped**.

Example 7: Stop a service on a remote system

This example stops a service on a remote computer. For more information, see Invoke-Command.

```
$Cred = Get-Credential
$S = Get-Service -Name Schedule
Invoke-Command -ComputerName server01.contoso.com -Credentia
  Set-Service -InputObject $S -Status Stopped
}
```

`Get-Credential` prompts for a username and password, and stores the credentials in the `$Cred` variable. `Get-Service` uses the **Name** parameter to specify the **Schedule** service. The object is stored in the variable, `$S`.

`Invoke-Command` uses the **ComputerName** parameter to specify a remote computer. The **Credential** parameter uses the `$Cred` variable to sign on to the computer. The **ScriptBlock** calls `Set-Service`. The **InputObject** parameter specifies the service object stored `$S`. The **Status** parameter sets the service to **Stopped**.

Example 8: Change credential of a service

This example changes the credentials that are used to manage a service.

```
$credential = Get-Credential
Set-Service -Name Schedule -Credential $credential
```

`Get-Credential` prompts for a username and password, and stores the credentials in the `$credential` variable. `Set-Service` uses the **Name** parameter to specify the **Schedule** service. The **Credential** parameter uses the `$credential` variable and updates the **Schedule** service.

Example 9: Change the SecurityDescriptor of a service

This example changes a service's **SecurityDescriptor**.

```
$SDDL = "D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOC
Set-Service -Name "BITS" -SecurityDescriptorSddl $SDDL
```

The **SecurityDescriptor** is stored in the `$SDDL` variable. `Set-Service` uses the **Name** parameter to specify the **BITS** service. The **SecurityDescriptorSddl** parameter uses `$SDDL` to change the **SecurityDescriptor** for the **BITS** service.

Example 10: Set the startup type for multiple services

The `Set-Service` cmdlet only accepts one service name at a time. However, you can pipe multiple services to `Set-Service` to change the configuration of multiple services.

```
Get-Service SQLWriter,spooler |
    Set-Service -StartupType Automatic -PassThru |
    Select-Object Name, StartType

Name      StartType
```

```
----      ---------
spooler   Automatic
SQLWriter Automatic
```

Parameters

`-Confirm`

Prompts you for confirmation before running `Set-Service`.

Expand table

Type:	SwitchParameter
Aliases:	cf
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Credential`

Specifies the account used by the service as the Service Logon Account.

Type a user name, such as **User01** or **Domain01\User01**, or enter a **PSCredential** object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, this cmdlet prompts you for a password.

Credentials are stored in a PSCredential object and the password is stored as a SecureString.

> ⓘ **Note**
>
> For more information about **SecureString** data protection, see How secure is SecureString?.

This parameter was introduced in PowerShell 6.0.

Expand table

Type:	PSCredential
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Description`

Specifies a new description for the service.

The service description appears in **Computer Management, Services**. The **Description** isn't a property of the `Get-Service` **ServiceController** object. To see the service description, use `Get-CimInstance` that returns a **Win32_Service** object that represents the service.

Expand table

Type:	String
Position:	Named
Default value:	None

Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-DisplayName`

Specifies a new display name for the service.

> **Note**
>
> Typically, `Set-Service` only operates on Windows services and not drivers. However, if you specify the name of a driver, `Set-Service` can target the driver.

Expand table

Type:	String
Aliases:	DN
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Force`

Specifies the Stop mode of the service. This parameter only works when `-Status Stopped` is used. If enabled, `Set-Service` stops the dependent services before the target service is stopped. By default, exceptions are raised when other running services depend on the target service.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-InputObject`

Specifies a **ServiceController** object that represents the service to change. Enter a variable that contains the object, or type a command or expression that gets the object, such as a `Get-Service` command. You can use the pipeline to send a service object to `Set-Service`.

Expand table

Type:	ServiceController
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	False

`-Name`

Specifies the service name of the service to be changed. Wildcard characters aren't permitted. You can use the pipeline to send a service name to `Set-Service`.

> ⓘ **Note**
>
> Typically, `Set-Service` only operates on Windows services and not drivers. However, if you specify the name of a driver, `Set-Service` can target the driver.

Expand table

Type:	String
Aliases:	ServiceName, SN
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	False

`-PassThru`

Returns a **ServiceController** object that represents the services that were changed. By default, `Set-Service` doesn't generate any output.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False

Accept wildcard characters:	False

`-SecurityDescriptorSddl`

Specifies the **SecurityDescriptor** for the service in **Sddl** format. The account calling `Set-Service` with this parameter must have the WRITE_DAC and WRITE_OWNER permissions. For more information, see Service security and access rights.

Expand table

Type:	String
Aliases:	sd
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-StartupType`

Specifies the start mode of the service.

The acceptable values for this parameter are as follows:

- **Automatic** - The service is started or was started by the operating system, at system start-up. If an automatically started service depends on a manually started service, the manually started service is also started automatically at system startup.
- **AutomaticDelayedStart** - Starts shortly after the system boots.

- **Disabled** - The service is disabled and cannot be started by a user or application.
- **InvalidValue** - Has no effect. The cmdlet does not return an error but the StartupType of the service is not changed.
- **Manual** - The service is started only manually, by a user, using the Service Control Manager, or by an application.

⛶ Expand table

Type:	ServiceStartupType
Aliases:	StartMode, SM, ST, StartType
Accepted values:	Automatic, AutomaticDelayedStart, Disabled, InvalidValue, Manual
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Status`

Specifies the status for the service.

The acceptable values for this parameter are as follows:

- **Paused**. Suspends the service.
- **Running**. Starts the service.
- **Stopped**. Stops the service.

⛶ Expand table

Type:	String

Accepted values:	Paused, Running, Stopped
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-WhatIf`

Shows what would happen if `Set-Service` runs. The cmdlet isn't run.

Expand table

Type:	SwitchParameter
Aliases:	wi
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

Inputs

ServiceController

You can pipe a service object to this cmdlet.

String

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

Outputs

None

By default, this cmdlet returns no output.

ServiceController

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

Notes

This cmdlet is only available on Windows platforms.

`Set-Service` requires elevated permissions. Use the **Run as administrator** option.

`Set-Service` can only control services when the current user has permissions to manage services. If a command doesn't work correctly, you might not have the required permissions.

To find a service's service name or display name, use `Get-Service`. The service names are in the **Name** column and the display names are in the **DisplayName** column.

Related Links

- Get-Service
- New-Service
- Restart-Service
- Resume-Service
- Start-Service
- Stop-Service

- Suspend-Service
- Remove-Service

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

PowerShell feedback

PowerShell is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback

English (United States)   Your Privacy Choices   Theme

Manage cookies   Previous Versions   Blog   Contribute   Privacy   Terms of Use   Trademarks