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

PowerShell Get-Help on command 'Start-Service'

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

NAME

Start-Service

SYNOPSIS

Starts one or more stopped services.

SYNTAX

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

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

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

DESCRIPTION

The `Start-Service` cmdlet sends a start message to the Windows Service Controller for each of the specified services. If a service is already running, the message is ignored without error. You can specify the services by their service names or display names, or you can use the InputObject parameter to supply a service object that represents the services that you want to start.

PARAMETERS

-DisplayName <System.String[]>

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

-InputObject <System.ServiceProcess.ServiceController[]>

Specifies ServiceController objects representing the services to be started. Enter a variable that contains the objects, or type a command or expression that gets the objects.

-Name <System.String[]>

Specifies the service names for the service to be started.

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

-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: Start a service by using its name ---------

Start-Service -Name "eventlog"

-- Example 2: Display information without starting a service --

Start-Service -DisplayName *remote* -WhatIf

The DisplayName parameter identifies the services by their display name
instead of their service name. The WhatIf parameter causes the cmdlet to
display what would happen when you run the command but does not make changes.
Example 3: Start a service and record the action in a text file

$s = Get-Service wmi
Start-Service -InputObject $s -PassThru | Format-List >> services.txt

First we use `Get-Service` to get an object that represent the WMI service and store it in the `$s` variable. Next, we start the service. Without the PassThru parameter, `Start-Service` does not create any output. The pipeline operator (`|`) passes the object output by `Start-Service` to the `Format-List` cmdlet to format the object as a list of its properties. The append redirection operator (`>>`) redirects the output to the services.txt file. The output is added to the end of the existing file.

---------- Example 4: Start a disabled service ----------

```
PS> Start-Service tlntsvr
Start-Service : Service 'Telnet (TlntSvr)' cannot be started due to the
following error: Cannot start service TlntSvr on computer '.'.
At line:1 char:14
+ Start-Service  <<<< tlntsvr

PS> Get-CimInstance win32_service | Where-Object Name -eq "tlntsvr"
ExitCode  : 0
Name      : TlntSvr
ProcessId : 0
StartMode : Disabled
State     : Stopped
Status    : OK

PS> Set-Service tlntsvr -StartupType manual
PS> Start-Service tlntsvr
```

The first attempt to start the Telnet service (tlntsvr) fails. The `Get-CimInstance` command shows that the StartMode property of the Tlntsvr service is Disabled . The `Set-Service` cmdlet changes the start type to Manual . Now, we can resubmit the `Start-Service` command. This time, the command succeeds. To verify that the command succeeded, run `Get-Service`.

REMARKS

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

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

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

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