

## System pomocy – info

Szczegółowo poznasz opcje poleceń Get-Help, Get-Command i Show-Command

Zobaczysz, jak przekazywać parametry przez nazwę i pozycję

Dowiesz się, jak aktualizować pomoc

Będziesz wiedział, co to tak zwane artykuły About

```
PS C:\Windows\system32> help Get-Service -ShowWindow
```

Różnią się ilością parametrów / [] - opcjonalny

### Syntax

```
Get-Service [-ComputerName <System.String[]> [-DependentServices ] -DisplayName <System.String[]> [-Exclude <System.String[]> [-Include <System.String[]>] [-RequiredServices ] [<CommonParameters>]
```

```
Get-Service [-ComputerName <System.String[]> [-DependentServices ] [-Exclude <System.String[]>] [-Include <System.String[]>] [-InputObject <System.ServiceProcess.ServiceController[]>] [-RequiredServices ] [<CommonParameters>]
```

```
Get-Service [-Name <System.String[]>] [-ComputerName <System.String[]>] [-DependentServices ] [-Exclude <System.String[]>] [-Include <System.String[]>] [-RequiredServices ] [<CommonParameters>]
```

```
PS C:\Windows\system32> Get-Service Spooler
```

Status	Name	DisplayName
Running	Spooler	Bufor wydruku

Jeśli opuszczasz nazwę parametru to przekazujesz ją przez pozycję. Musisz pamiętać w jakiej kolejności należy pisać parametry

```
PS C:\Windows\system32> Get-Service -Name Spooler
```

Status	Name	DisplayName
Running	Spooler	Bufor wydruku

Przekazywany parametr przez nazwę, nie musisz pamiętać kolejności parametrów

```
PS C:\Windows\system32> Get-Service -Name Spooler -ComputerName localhost

Status   Name      DisplayName
-----
Running  Spooler   Bufor wydruku
```

ComputerName localhost: Wskazuje, że chcemy sprawdzić tę

usługę na lokalnym komputerze (o nazwie “localhost”).

```
PS C:\Windows\system32> Get-Service Spooler localhost
Get-Service : A positional parameter cannot be found that accepts argument 'localhost'.
At line:1 char:1
+ Get-Service Spooler localhost
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Get-Service], ParameterBindingException
+ FullyQualifiedErrorId : PositionalParameterNotFound,Microsoft.PowerShell.Commands.GetServiceCommand
```

Zaznaczony -ComputerName nie jest w nawiasie kwadratowym więc jest **obowiązkowe**

#### Syntax

```
Get-Service [-ComputerName <System.String[]>] [-DependentServices ] [-DisplayName <System.String[]>] [-Exclude <System.String[]>] [-Include <System.String[]>] [-RequiredServices ] [<CommonParameters>]
```

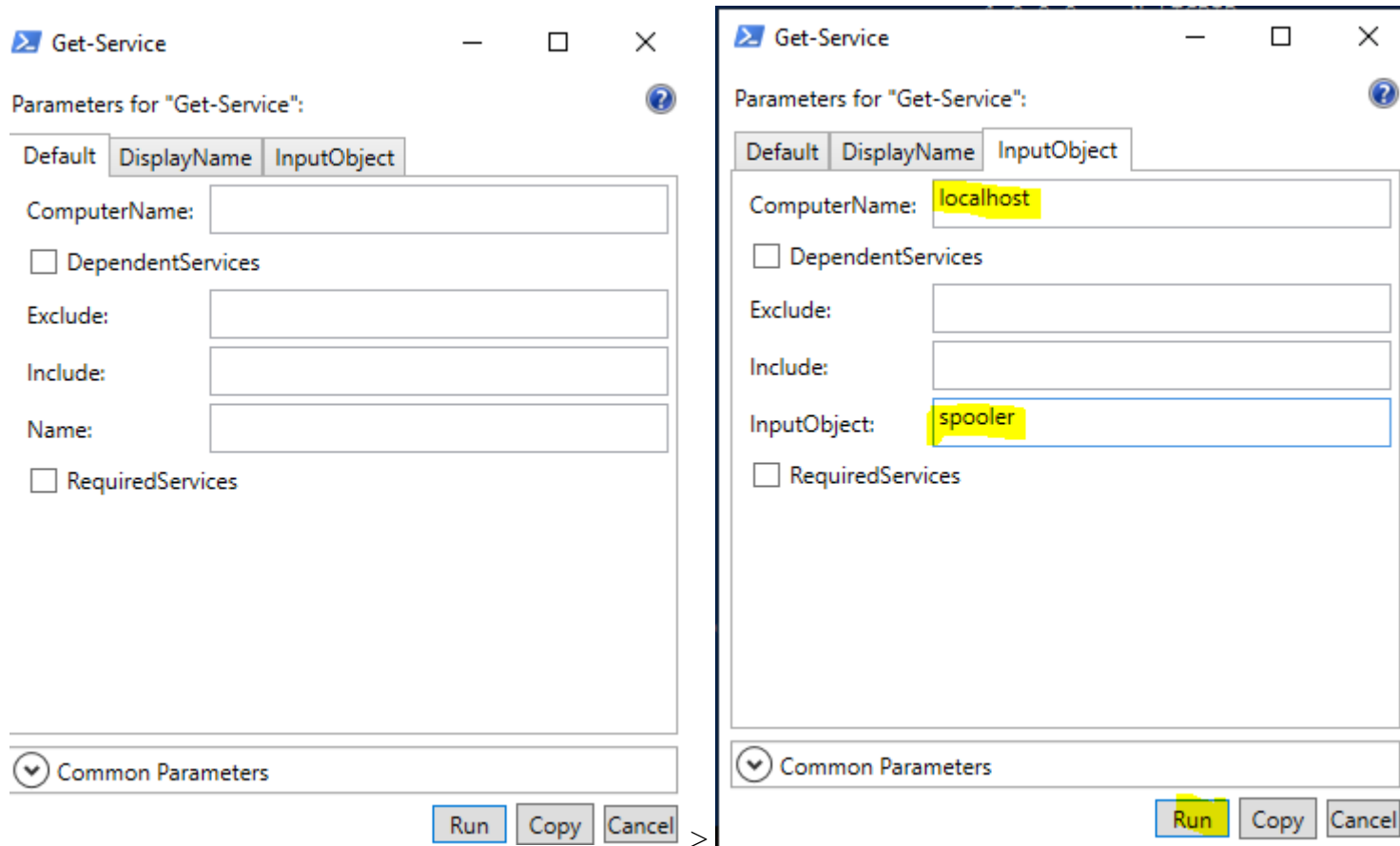
```
Get-Service [-ComputerName <System.String[]>] [-DependentServices ] [-Exclude <System.String[]>] [-Include <System.String[]>] [-InputObject <System.ServiceProcess.ServiceController[]>] [-RequiredServices ] [<CommonParameters>]
```

```
Get-Service [[-Name] <System.String[]>] [-ComputerName <System.String[]>] [-DependentServices ] [-Exclude <System.String[]>] [-Include <System.String[]>] [-RequiredServices ] [<CommonParameters>]
```

Polecenie poniżej pozwala na wygodne wpisanie parametrów

```
PS C:\Windows\system32> Show-Command Get-Service
```

Okno poniżej ma zmienną ilość zakładek – chroni przed pomieszaniem parametrów



Efekt

```
PS C:\Windows\system32> Get-Service -ComputerName localhost -InputObject spooler
```

Status	Name	DisplayName
Running	spooler	Bufor wydruku

```
PS C:\Windows\system32> help Get-Service -full
```

## NAME

Get-Service

## SYNOPSIS

Gets the services on a local or remote computer.

## SYNTAX

```
Get-Service [-ComputerName <System.String[]>] [-DependentServices] -DisplayName <System.String[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-RequiredServices] [<CommonParameters>]
```

```
Get-Service [-ComputerName <System.String[]>] [-DependentServices] [-Exclude <System.String[]>] [-Include <System.ServiceProcess.ServiceController[]>] [-RequiredServices] [<CommonParameters>]
```

```
Get-Service [[-Name] <System.String[]>] [-ComputerName <System.String[]>] [-DependentServices] [-Exclude <System.String[]>] [-Include <System.String[]>] [-RequiredServices] [<CommonParameters>]
```

## DESCRIPTION

The `Get-Service` cmdlet gets objects that represent the services on a local computer or on a remote computer, including stopped services. By default, when `Get-Service` is run without parameters, all the local computer's services are returned.

You can direct this cmdlet to get only particular services by specifying the service name or the display name of the service. You can pipe service objects to this cmdlet.

## PARAMETERS

`-ComputerName <System.String[]>`

Gets the services running on the specified computers. The default is the local computer.

Type the NetBIOS name, an IP address, or a fully qualified domain name (FQDN) of a remote computer. To specify the local computer, type the computer name, a dot (`.`), or `localhost`.

This parameter does not rely on Windows PowerShell remoting. You can use the `ComputerName` parameter of `Get-Service` if the remote computer is not configured to run remote commands.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

`-DependentServices <System.Management.Automation.SwitchParameter>`

Indicates that this cmdlet gets only the services that depend upon the specified service.

Required?	false
-----------	-------

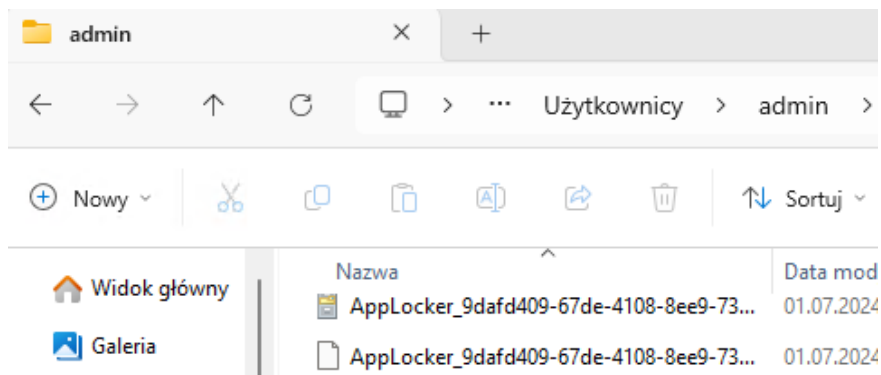
```
----- Example 9: Get services on multiple computers -----  
  
Get-Service -Name "WinRM" -ComputerName "localhost", "Server01", "Server02" |  
  Format-Table -Property MachineName, Status, Name, DisplayName -auto  
  
MachineName    Status    Name    DisplayName  
-----  
localhost      Running  WinRM   Windows Remote Management (WS-Management)  
Server01       Running  WinRM   Windows Remote Management (WS-Management)  
Server02       Running  WinRM   Windows Remote Management (WS-Management)  
  
This command uses the `Get-Service` cmdlet to run a `Get-Service Winrm` command on two remote computers and the local computer (`localhost`).  
  
The command runs on the remote computers, and the results are returned to the local computer. A pipeline operator (`|`) sends the results to the `Format-Table` cmdlet, which formats the services as a table. The `Format-Table` command uses the Property parameter to specify the properties displayed in the table, including the MachineName property.  
----- Example 10: Get the dependent services of a service -----  
  
Get-Service "WinRM" -RequiredServices  
  
█  
--- Example 11: Get a service through the pipeline operator ---  
  
"WinRM" | Get-Service
```

```
PS C:\Windows\system32> Update-Help
```

```
Updating Help for module iSCSI  
Locating Help Content...  
[
```

```
PS C:\Windows\system32> Update-Help -force - wymuszenie aktualizacji
```

```
PS C:\Windows\system32> Save-Help -DestinationPath C:\users\admin\ - pobranie aktualizacji do określonego folderu
```



Wymuszenie zainstalowania pobranych modułów

```
PS C:\Windows\system32> Update-Help -SourcePath C:\users\admin\
```

Sprawdź jakie artykuły pomocy są dostępne

```
PS C:\Windows\system32> help about*
```

Poniższy przedstawia info o zmiennych środowiskowych

```
about_Environment_Provider      HelpFile
about_Environment_Variables     HelpFile
about_Eventlogs                 HelpFile
```

Dokładna specyfikacja zmiennych

```
PS C:\Windows\system32> help about_Environment_Variables
```

```
PS C:\Windows\system32> help about_Environment_Variables -ShowWindow
```

Find:

Previous

Next

## ABOUT\_ENVIRONMENT\_VARIABLES

### Short description

Describes how to access and manage environment variables in PowerShell.

```
PS C:\Windows\system32> help *restart*
```

Name	Category	Module	Synopsis
Restart-Computer	Cmdlet	Microsoft.PowerShell.M...	Restarts the operating system on local and remote computers.
Restart-Service	Cmdlet	Microsoft.PowerShell.M...	Stops and then starts one or more services.
Restart-NetAdapter	Function	NetAdapter	...
Restart-PcsvDevice	Function	PcsvDevice	...
Restart-PrintJob	Function	PrintManagement	...

```
PS C:\Windows\system32> help *beep*
```

## ABOUT\_SPECIAL\_CHARACTERS

### Short description

Describes the special character sequences that control how PowerShell interprets the next characters in the sequence.

Można przeczytać jak wydobyć dźwięk z PowerShell

## Long description

PowerShell supports a set of special character sequences that are used to represent characters that aren't part of the standard character set. The sequences are commonly known as `_escape sequences_`.

Escape sequences begin with the backtick character, known as the grave accent (ASCII 96), and are case-sensitive. The backtick character can also be referred to as the `_escape character_`.

Escape sequences are only interpreted when contained in double-quoted (") strings.

PowerShell recognizes these escape sequences:

Sequence	Description
<code>`0</code>	Null
<code>`a</code>	Alert
<code>`b</code>	Backspace
<code>`f</code>	Form feed
<code>`n</code>	New line
<code>`r</code>	Carriage return
<code>`t</code>	Horizontal tab
<code>`v</code>	Vertical tab

Szczegółowo przyjrzałeś się `Get-Help`, `Get-Command` i `Show-Command`

Zobaczyłeś, jak przekazywać parametry przez nazwę i pozycję

Dowiedziałeś się, jak aktualizować pomoc

Wiesz, co to tak zwane artykuły About