

Pomoc na temat komend – info

Poznasz komendę Get-Help

Dowiesz się, jak odnaleźć właściwe polecenie za pomocą Get-Command

Podstawowe informacje o poleceniu Get-Service: Aby wyświetlić ogólne informacje o poleceniu Get-Service, wpisz w konsoli: **Get-Help Get-Service**

```
PS C:\Users\admin> get-help Get-Service
```

Pomoc jest podzielona na sekcje

```
NAME
    Get-Service

SYNTAX
    Get-Service [[-Name] <string[]>] [-ComputerName <string[]>] [-DependentServices] [-RequiredServices] [-Include <string[]>] [-Exclude <string[]>] [<CommonParameters>]

    Get-Service -DisplayName <string[]> [-ComputerName <string[]>] [-DependentServices] [-RequiredServices] [-Include <string[]>] [-Exclude <string[]>] [<CommonParameters>]

    Get-Service [-ComputerName <string[]>] [-DependentServices] [-RequiredServices] [-Include <string[]>] [-Exclude <string[]>] [-InputObject <ServiceController[]>] [<CommonParameters>]

ALIASES
    gsv

REMARKS
    Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only partial help.
    -- To download and install Help files for the module that includes this cmdlet, use Update-Help.
    -- To view the Help topic for this cmdlet online, type: "Get-Help Get-Service -Online" or
    go to https://go.microsoft.com/fwlink/?LinkID=113332.
```

Alias do get-help to help i man

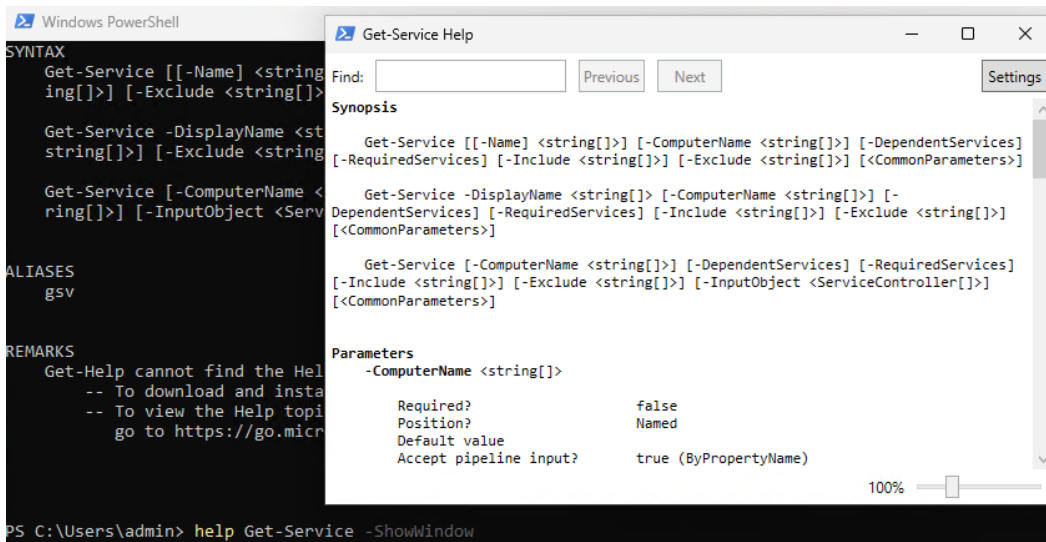
```
PS C:\Users\admin> help Get-Service
```

Jak widać mamy włączone stronicowanie

```
ALIASES
    gsv
    -- More --
```

Enter - przesunięta linijka w dół

spacja - przesunięty cały ekran

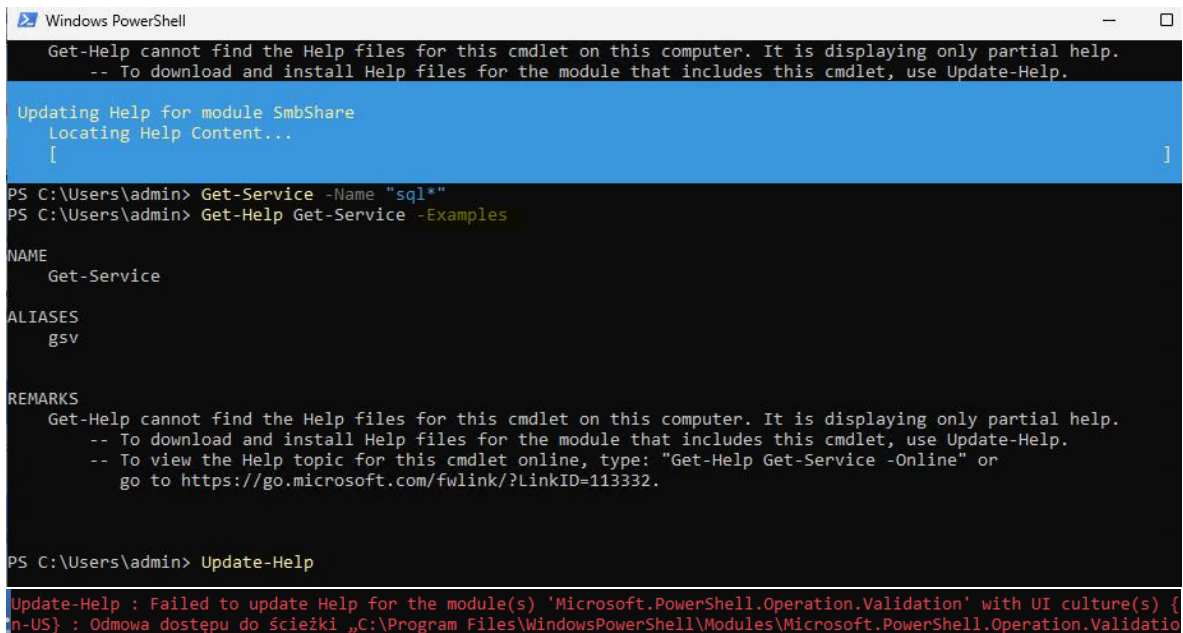


Filtrowanie wyników: Możesz również filtrować wyniki, na przykład wyświetlając tylko usługi, których nazwa zaczyna się od określonego ciągu znaków: `Get-Service -Name "sql*"`

```
PS C:\Windows\system32> Get-Service -Name "sql*"
PS C:\Windows\system32>
```

Powershell jest zbudowany z modułów do każdego z nich jest help i każdy można aktualizować.

Update-Help



Po aktualizacji z poziomu Administratora (`system32`)

```

Administrator: Windows PowerShell
PS C:\Windows\system32> help Get-Service -ShowWindow
Get-Service Help
Find:  Previous Next
Examples
----- Example 1: Get all services on the computer -----
Get-Service

--- Example 2: Get services that begin with a search string ---
Get-Service "wmi*"

--- Example 3: Display services that include a search string ---
Get-Service -Displayname "*network*"

Example 4: Get services that begin with a search string and an exclusion
Get-Service -Name "win*" -Exclude "WinRM"

---- Example 5: Display services that are currently active ----
Get-Service | Where-Object {$_.Status -eq "Running"}

```

Przykłady użycia polecenia Get-Service: Aby zobaczyć przykłady korzystania z polecenia Get-Service, użyj parametru -Examples: **Get-Help Get-Service -Examples**

```

----- 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> Get-Help Get-Service -Examples_

```

Pełne informacje o poleceniu Get-Service: Jeśli potrzebujesz bardziej szczegółowych informacji, użyj parametru -Full: **Get-Help Get-Service -Full**

```

RELATED LINKS
  Online Version: https://learn.microsoft.com/powershell/
  rshell-5.1&WT.mc_id=ps-gethelp
  New-Service
  Restart-Service
  Resume-Service
  Set-Service
  Start-Service
  Stop-Service
  Suspend-Service

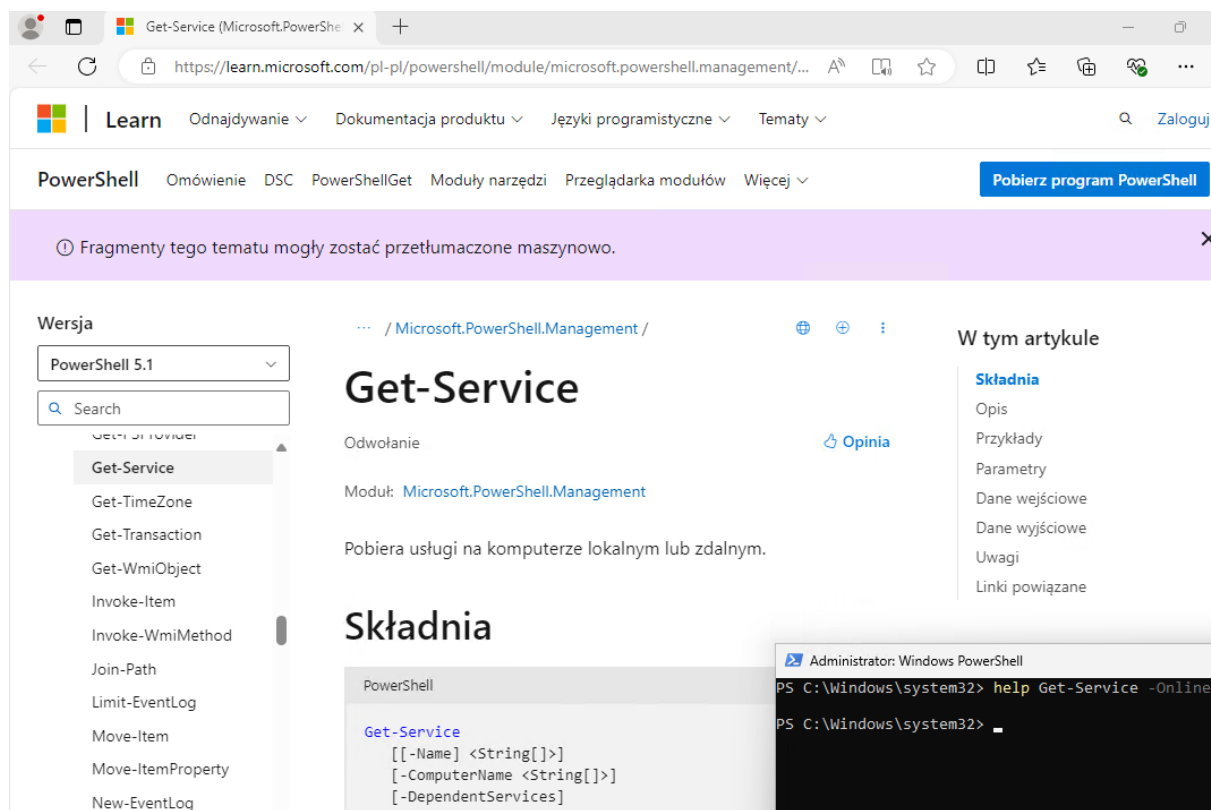
PS C:\Windows\system32> Get-Help Get-Service -Full_

```

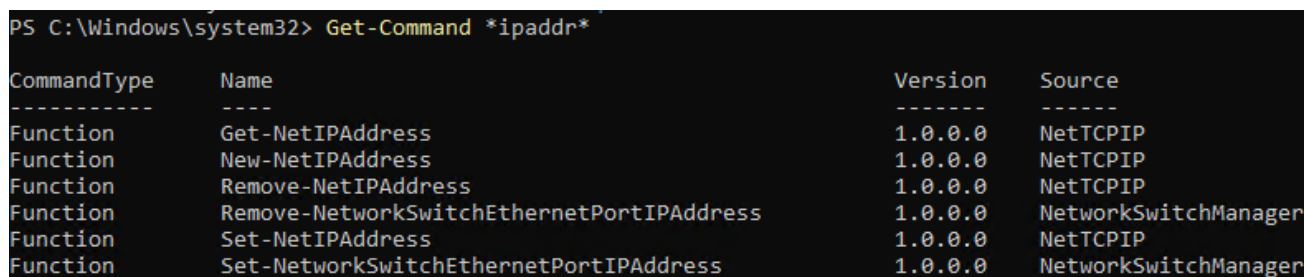
Alias clear w PowerShell jest używany jako skrót do polecenia Clear-Host. Oba te polecenia służą do wyczyszczenia konsoli, czyli ekranu wyjścia. Działają podobnie jak polecenie cls w wierszu poleceń

systemu Windows. aliasy są używane dla wygody, a wybór między clear a Clear-Host zależy od preferencji i kontekstu. **clear** alias to **cls**

Polecenie **help Get-Service -Online** służy do uzyskania dostępu do dokumentacji online dla polecenia Get-Service w PowerShellu. Wykonując to polecenie, otworzy się przeglądarka internetowa, a użytkownik zostanie przekierowany na stronę z oficjalną dokumentacją Microsoftu.



Polecenie **Get-Command** w PowerShellu służy do wyświetlania informacji o dostępnych poleceniach. Jeśli użyjesz **Get-Command *ipaddr***, otrzymasz listę wszystkich poleceń, których nazwa zawiera ciąg "ipaddr". To może obejmować różne polecenia związane z IP, ale nie jest to specyficzne dla jednego konkretnego zadania.



Polecenie **Get-NetIPAddress** w PowerShellu służy do uzyskiwania informacji o konfiguracji adresów IP, takich jak adresy IPv4, IPv6 oraz interfejsy IP, z którymi adresy są skojarzone. Bez podawania parametrów, to polecenie zwraca pełną konfigurację adresów IP dla komputera.

```

PS C:\Windows\system32> Get-Help Get-NetIPAddress

NAME
    Get-NetIPAddress

SYNOPSIS
    Gets the IP address configuration.

SYNTAX
    Get-NetIPAddress [[-IPAddress] <String[]>] [-AddressFamily {IPv4 | IPv6}] [-AddressState {Invalid | Tentative | Duplicate | Deprecated | Preferred}] [-AsJob] [-AssociatedIPInterface <CimInstance>] [-CimSession <CimSession[]>] [-IncludeAllCompartments] [-InterfaceAlias <String[]>] [-InterfaceIndex <UInt32[]>] [-PolicyStore <String>] [-PreferredLifetime <TimeSpan[]>] [-PrefixLength <Byte[]>] [-PrefixOrigin {Other | Manual | WellKnown | Dhcp | RouterAdvertisement}] [-SkipAsSource <Boolean[]>] [-SuffixOrigin {Other | Manual | WellKnown | Dhcp | Link | Random}] [-ThrottleLimit <Int32>] [-Type {Unicast | Anycast}] [-ValidLifetime <TimeSpan[]>] [<CommonParameters>]

DESCRIPTION
    The Get-NetIPAddress cmdlet gets the IP address configuration, such as IPv4 addresses, IPv6 addresses and the IP interfaces with which addresses are associated. Without parameters, this cmdlet gets the entire IP address configuration for the computer.

RELATED LINKS
    Online Version: https://learn.microsoft.com/powershell/module/nettcpip/get-netipaddress?view=windowsserver2022-ps&t.mc\_id=ps-gethelp
    Get-NetRoute
    New-NetIPAddress
    Remove-NetIPAddress
    Set-NetIPAddress

REMARKS
    To see the examples, type: "get-help Get-NetIPAddress -examples".
    For more information, type: "get-help Get-NetIPAddress -detailed".
    For technical information, type: "get-help Get-NetIPAddress -full".
    For online help, type: "get-help Get-NetIPAddress -online"

```

Kilka przykładów:

Pobieranie konfiguracji adresów IPv6: Aby uzyskać informacje o konfiguracji adresów IP dla wszystkich adresów IPv6 na komputerze, wykonaj polecenie:

```
Get-NetIPAddress -AddressFamily IPv6
```

Pobieranie informacji o adresach IP i formatowanie wyniku: Polecenie:

```
Get-NetIPAddress | Format-Table
```

zwraca informacje o konfiguracji adresów IP i wyświetla je w formie tabeli. Ten format jest wygodny, gdy komputer ma wiele adresów IP.

Pobieranie informacji o adresach IP na podstawie indeksu interfejsu: Jeśli chcesz uzyskać informacje o konfiguracji adresów IP dla określonego indeksu interfejsu (np. indeks 12), użyj:

```
Get-NetIPAddress -InterfaceIndex 12
```

Sortowanie informacji o adresach IP według indeksu interfejsu: Polecenie:

```
Get-NetIPAddress | Sort-Object -Property InterfaceIndex | Format-Table
```

sortuje informacje o adresach IP numerycznie według indeksu interfejsu i wyświetla je w formie tabeli. To ułatwia znalezienie informacji o adresach IP na podstawie indeksu interfejsu.

Filtrowanie informacji o adresach IP: Aby uzyskać konfigurację adresów IP dla adresów, których ValidLifetime jest krótszy niż jeden dzień, użyj:

```
Get-NetIPAddress | Where-Object { $_.ValidLifetime -Lt ([TimeSpan]::FromDays(1)) }
```

Możesz również użyć podobnego polecenia, aby uzyskać informacje o adresach IP z nieskończoną ValidLifetime:

```
Get-NetIPAddress | Where-Object { $_.ValidLifetime -Eq ([TimeSpan]::MaxValue) }
```

Polecenie `Get-Command -noun *ipaddr*` w Windows PowerShell służy do wyszukiwania komend związanych z frazą “ipaddr”

```
PS C:\Windows\system32> Get-Command -noun *ipaddr*

CommandType      Name                                     Version      Source
-----
Function         Get-NetIPAddress                       1.0.0.0     NetTCPIP
Function         New-NetIPAddress                       1.0.0.0     NetTCPIP
Function         Remove-NetIPAddress                   1.0.0.0     NetTCPIP
Function         Remove-NetworkSwitchEthernetPortIPAd 1.0.0.0     NetworkSwitchManager
Function         Set-NetIPAddress                       1.0.0.0     NetTCPIP
Function         Set-NetworkSwitchEthernetPortIPAdre 1.0.0.0     NetworkSwitchManager
```

Ciekawostki:

Publiczny (zewnętrzny) adres IP: Aby poznać swój publiczny adres IP, użyj jednego z poniższych poleceń: `curl ifcfg.me`

To polecenie natychmiast wyświetli twój obecny publiczny adres IP w wierszu poleceń. Działa poprzez pobranie zawartości strony ifconfig.me, która jest bardzo prostą stroną internetową, wyświetlającą tylko twój publiczny adres IP.

Alternatywnie, możesz użyć innych metod, takich jak:

```
curl icanhazip.com
```

lub

```
nslookup myip.opendns.com
```

Wyjaśnienie, jak używać znaków wieloznacznych (wildcard) w programie PowerShell.

```
PS C:\Windows\system32> Get-Command *wild*
PS C:\Windows\system32> help *wild*
```

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

ABOUT_WILDCARDS

Short description
Describes how to use wildcard characters in PowerShell.

Long description
Wildcard characters represent one or many characters. You can use them to
create word patterns in commands. Wildcard expressions are used with the
-like operator or with any parameter that accepts wildcards.

For example, to match all the files in the C:\Techdocs directory with a
.ppt file name extension, type:

    Get-ChildItem C:\Techdocs\*.ppt

In this case, the asterisk (*) wildcard character represents any characters
that appear before the .ppt file name extension.

Wildcard expressions are simpler than regular expressions. For more
information, see about_Regular_Expressions.

PowerShell supports the following wildcard characters:
```

To co można z tego wywnioskować to przykłady znaków wieloznacznych w programie PowerShell:

1. Asterystka (*) - Dopasowanie zero lub więcej znaków:

- o a* dopasuje aA, ag i Apple.
- o a* nie dopasuje banana.

2. Znak zapytania (?) - Dopasowanie jednego znaku:

- o ?n dopasuje an, in i on.
- o ?n nie dopasuje ran.

3. Znak zapytania (?) - Dopasowanie zero lub jednego znaku (dla plików i katalogów):

- o ?.txt dopasuje a.txt i b.txt.
- o ?.txt nie dopasuje ab.txt.

4. Nawiasy kwadratowe ([]) - Dopasowanie zakresu znaków:

- o [a-l]ook dopasuje book, cook i look.
- o [a-l]ook nie dopasuje took.

5. Nawiasy kwadratowe ([]) - Dopasowanie konkretnych znaków:

- o [bc]ook dopasuje book i cook.
- o [bc]ook nie dopasuje hook.

6. Backtick (`) - Dopasowanie znaku jako dosłownego (nie jako znaku wieloznacznego):

- o 124`dopasuje124`.
- o 12*4nie dopasuje1234`.

Możesz również używać wielu znaków wieloznacznych w tym samym wzorcu słów. Na przykład, aby znaleźć pliki tekstowe o nazwach zaczynających się od liter A do L, wpisz:

```
Get-ChildItem C:\Techdocs\[a-l]*.txt
```

Warto zauważyć, że dopasowywanie znaków wieloznacznych dla elementów systemu plików działa inaczej niż dla ciągów znaków. Jeśli potrzebujesz więcej informacji, zobacz sekcję “Remarks” w dokumentacji metody `DirectoryInfo.GetFiles(String, EnumerationOptions)`.

Czasami może być potrzebne dopasowanie dosłownego znaku, a nie traktowanie go jako znaku wieloznacznego. W takich przypadkach możesz użyć znaku backtick (`), aby uniknąć interpretacji znaku jako wieloznacznego. Na przykład `*hello?*` dopasuje ciągi zawierające “hello?”.

Wiele cmdletów akceptuje znaki wieloznaczne w wartościach parametrów. Temat Pomocy dla każdego cmdletu opisuje, które parametry akceptują znaki wieloznaczne. Użycie znaków wieloznacznych w parametrach jest nieczułe na wielkość liter.

Możesz również używać znaków wieloznacznych w poleceniach i blokach skryptu, na przykład do tworzenia wzorca słów reprezentującego wartości właściwości. Przykładowo, poniższe polecenie pobiera usługi, w których wartość właściwości `SERVICETYPE` zawiera “INTERACTIVE”:


```
Get-Service | Where-Object {$_.ServiceType -Like "*Interactive*"}
```

W drugim przykładzie warunek w instrukcji If używa znaków wieloznacznych do znalezienia wartości właściwości. Jeśli opis punktu przywracania zawiera "POWERSHELL", polecenie dodaje wartość właściwości CREATIONTIME punktu przywracania do pliku dziennika.

```
$p = Get-ComputerRestorePoint  
foreach ($point in $p) {  
    if ($point.description -like "*Power
```

Gdy chcesz dowiedzieć się coś więcej o module, który masz.

Polecenie **Get-Command -Module NetTCPIP** w programie PowerShell służy do wyświetlania listy dostępnych poleceń związanych z modułem NetTCPIP. Ten moduł zawiera polecenia związane z konfiguracją protokołu TCP/IP.

```
PS C:\Windows\system32> Get-Command -Module NetTCPIP
```

CommandType	Name	Version	Source
Function	Find-NetRoute	1.0.0.0	NetTCPIP
Function	Get-NetCompartment	1.0.0.0	NetTCPIP
Function	Get-NetIPAddress	1.0.0.0	NetTCPIP
Function	Get-NetIPConfiguration	1.0.0.0	NetTCPIP
Function	Get-NetIPInterface	1.0.0.0	NetTCPIP
Function	Get-NetIPv4Protocol	1.0.0.0	NetTCPIP
Function	Get-NetIPv6Protocol	1.0.0.0	NetTCPIP
Function	Get-NetNeighbor	1.0.0.0	NetTCPIP
Function	Get-NetOffloadGlobalSetting	1.0.0.0	NetTCPIP
Function	Get-NetPrefixPolicy	1.0.0.0	NetTCPIP
Function	Get-NetRoute	1.0.0.0	NetTCPIP
Function	Get-NetTCPConnection	1.0.0.0	NetTCPIP
Function	Get-NetTCPSetting	1.0.0.0	NetTCPIP
Function	Get-NetTransportFilter	1.0.0.0	NetTCPIP
Function	Get-NetUDPEndpoint	1.0.0.0	NetTCPIP
Function	Get-NetUDPSetting	1.0.0.0	NetTCPIP
Function	New-NetIPAddress	1.0.0.0	NetTCPIP
Function	New-NetNeighbor	1.0.0.0	NetTCPIP
Function	New-NetRoute	1.0.0.0	NetTCPIP
Function	New-NetTransportFilter	1.0.0.0	NetTCPIP
Function	Remove-NetIPAddress	1.0.0.0	NetTCPIP
Function	Remove-NetNeighbor	1.0.0.0	NetTCPIP
Function	Remove-NetRoute	1.0.0.0	NetTCPIP
Function	Remove-NetTransportFilter	1.0.0.0	NetTCPIP
Function	Set-NetIPAddress	1.0.0.0	NetTCPIP
Function	Set-NetIPInterface	1.0.0.0	NetTCPIP
Function	Set-NetIPv4Protocol	1.0.0.0	NetTCPIP
Function	Set-NetIPv6Protocol	1.0.0.0	NetTCPIP
Function	Set-NetNeighbor	1.0.0.0	NetTCPIP
Function	Set-NetOffloadGlobalSetting	1.0.0.0	NetTCPIP
Function	Set-NetRoute	1.0.0.0	NetTCPIP
Function	Set-NetTCPSetting	1.0.0.0	NetTCPIP
Function	Set-NetUDPSetting	1.0.0.0	NetTCPIP
Function	Test-NetConnection	1.0.0.0	NetTCPIP

Przykłady poleceń z tego modułu:

1. **Get-NetTCPConnection:** Polecenie Get-NetTCPConnection zwraca informacje o bieżących połączeniach TCP. Możesz użyć go, aby wyświetlić właściwości połączenia, takie jak lokalny lub zdalny adres IP, lokalny lub zdalny port oraz stan połączenia.
2. **Get-NetIPAddress:** Polecenie Get-NetIPAddress pozwala uzyskać konfigurację adresu IP. Wyświetla informacje o adresach IP przypisanych do interfejsów sieciowych na twoim komputerze.
3. **Find-NetRoute:** Polecenie Find-NetRoute znajduje najlepszy lokalny adres IP i najlepszą trasę do osiągnięcia zdalnego adresu. Pomaga w analizie tras sieciowych.

Poznałeś komendę Get-Help

Dowiedziałeś się, jak odnaleźć właściwe polecenie za pomocą Get-Command