



python



PowerShell

FPDF Library
PDF generator

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

PowerShell Get-Help on command 'Get-NetRoute'

PS C:\Users\wahid> Get-Help Get-NetRoute

NAME

Get-NetRoute

SYNOPSIS

Gets the IP route information from the IP routing table.

SYNTAX

```
Get-NetRoute [[-DestinationPrefix] <String[]>] [-AddressFamily {IPv4 | IPv6}]  
[-AsJob] [-AssociatedIPInterface <CimInstance>] [-CimSession <CimSession[]>]  
[-CompartmentId <UInt32[]>] [-IncludeAllCompartments] [-InterfaceAlias  
<String[]>] [-InterfaceIndex <UInt32[]>] [-InterfaceMetric <UInt32[]>]  
[-NextHop <String[]>] [-PolicyStore <String>] [-PreferredLifetime  
<TimeSpan[]>] [-Protocol {Other | Local | NetMgmt | Icmp | Egp | Ggp | Hello |  
Rip | Isis | Eisis | Igrp | Bbn | Ospf | Bgp | Idpr | Eigrp | Dvmrp | Rpl |  
Dhcp}] [-Publish {No | Age | Yes}] [-RouteMetric <UInt16[]>] [-State  
<State[]>] [-ThrottleLimit <Int32>] [-ValidLifetime <TimeSpan[]>]  
[<CommonParameters>]
```

DESCRIPTION

The Get-NetRoute cmdlet gets IP route information from the IP routing table, including destination network prefixes, next hop IP addresses, and route metrics. Run this cmdlet without any parameters to get all IP routes from the routing table. Specify parameters to narrow your results. For instance, you can specify a particular interface or an IP address family.

For more information about routing, see Chapter 5 - IP Routing (<https://technet.microsoft.com/library/bb727001.aspx>) in the TechNet library.

PARAMETERS

-AddressFamily <AddressFamily[]>

Specifies an IP address family. The cmdlet gets IP routes of the families that you specify. The acceptable values for this parameter are:

- IPv4

- IPv6

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete.

-AssociatedIPInterface <CimInstance>

Specifies an IP interface as a CIM object. The cmdlet gets IP routes that belong to the interface that you specify. To obtain an IP interface, use the Get-NetIPInterface cmdlet.

-CimSession <CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a New-CimSession (<https://go.microsoft.com/fwlink/p/?LinkId=227967>) or [Get-CimSession](<https://go.microsoft.com/fwlink/p/?LinkId=227966>)cmdlet.

The default is the current session on the local computer.

-CompartmentId <UInt32[]>

Specifies an array of identifiers for network compartments in the protocol stack. By default, the cmdlet only gets Net routes in the default compartment. If you specify a value, the cmdlet gets any matching Net routes in all compartments in this field.

-DestinationPrefix <String[]>

Specifies an array of destination prefixes of IP routes. The cmdlet gets IP routes that use the prefixes that you specify. A destination prefix contains an IP address prefix and a prefix length, separated by a slash (/). A value of 0.0.0.0/0 for IPv4 or ::/0 for IPv6 indicates that the value of the NextHop parameter is a default gateway.

-IncludeAllCompartments [<SwitchParameter>]

Indicates that the cmdlet includes routes from all configured network compartments. If you do not specify this parameter, the cmdlet gets only routes in the default network compartment.

-InterfaceAlias <String[]>

Specifies an array of aliases of network interfaces. The cmdlet gets IP routes for the interfaces that have the aliases that you specify.

-InterfaceIndex <UInt32[]>

Specifies an array of indexes of network interfaces. The cmdlet gets IP routes for the interfaces located at the indexes that you specify.

-InterfaceMetric <UInt32[]>

Specifies an array of integer interface metrics for network interfaces. The cmdlet gets IP routes for the interfaces that have the metric that you specify.

-NextHop <String[]>

Specifies an array of next hop values. The cmdlet gets IP routes that have the next hop values that you specify. A value of 0.0.0.0 for IPv4 or :: for IPv6 indicates that the route is on the local subnet.

-PolicyStore <String>

Specifies a PolicyStore value. The cmdlet gets IP routes that have the PolicyStore value that you specify. The acceptable values for this parameter are:

- ActiveStore. The IP address information is valid. - PersistentStore. The computer saves IP address information across restarts. When the computer restarts, it copies the saved settings to the ActiveStore.

-PreferredLifetime <TimeSpan[]>

Specifies an array of values for preferred lifetime, as TimeSpan objects, of IP routes. The cmdlet gets entries that have these values. To obtain a TimeSpan object, use the New-TimeSpan cmdlet. For more information, type ``Get-Help New-TimeSpan``.

-Protocol <Protocol[]>

Specifies an array of types of routing protocols. The cmdlet gets entries that use the protocols that you specify. The acceptable values for this parameter are:

- Bbn

- Bgp

- Dhcp

- Dvmrp

- Egp

- Eigrp

- Esls

- Ggp

- Hello

- Icmp

- Idpr

- Igrp

- Isls

- Local

- NetMgmt

- Ospf

- Rip

- Rpl

- Other

-Publish <Publish[]>

Specifies an array of publish settings of IP routes. The cmdlet gets entries that have the publish values that you specify. The acceptable

values for this parameter are:

- No. Do not publish or advertise IP route information in router advertisements. - Yes. Publish and advertise IP route information with an infinite valid lifetime in router advertisements - Age. Publish and advertise IP route information with a finite valid lifetime in router advertisements.

-RouteMetric <UInt16[]>

Specifies an array of integer route metrics for IP routes. The cmdlet gets entries that have the metrics that you specify. To choose among multiple routes, the computer adds this value to the interface metric value. The computer selects the route with the lowest combined value. To modify the interface metric, use the Set-NetIPInterface cmdlet.

-State <State[]>

Specifies an array of state values for IP routes. The cmdlet gets entries that have the state values that you specify.

-ThrottleLimit <Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then Windows PowerShell® calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit applies only to the current cmdlet, not to the session or to the computer.

-ValidLifetime <TimeSpan[]>

Specifies an array of values for valid lifetimes, as TimeSpan objects, for IP routes. The cmdlet gets entries that have the lifetimes that you specify. To obtain a TimeSpan object, use the New-Timespan cmdlet.

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [about_CommonParameters \(https://go.microsoft.com/fwlink/?LinkID=113216\)](https://go.microsoft.com/fwlink/?LinkID=113216).

----- Example 1: Get all routes -----

```
PS C:\>Get-NetRoute | Format-List -Property *
```

This command gets all the routes for the computer, and then passes them to the Format-List cmdlet by using the pipeline operator. The Format-List cmdlet displays all the properties of an object. For more information, type ``Get-Help Format-List``.

----- Example 2: Get all IPv6 routes -----

```
PS C:\>Get-NetRoute -AddressFamily IPv6
```

This command gets the routes that belong to the IPv6 address family.

----- Example 3: Get routes for a specified interface -----

```
PS C:\>Get-NetRoute -InterfaceIndex 12
```

This command gets the IP routes associated with the interface that has an index of 12.

----- Example 4: Get the next hop for the default route -----

```
PS C:\>Get-NetRoute -DestinationPrefix "0.0.0.0/0" | Select-Object  
-ExpandProperty "NextHop"
```

This command gets the next hop for the default route. The next hop gateway for the default route is also known as the default gateway. The command gets the default IP routes, and passes them to the Select-Object cmdlet. That cmdlet displays the NextHop property for each default route. For more information

about displaying selected properties, type ``Get-Help Select-Object``.

----- Example 5: Get IP routes to non-local destinations -----

```
PS C:\>Get-NetRoute | Where-Object -FilterScript { $_.NextHop -Ne "::" } |  
Where-Object -FilterScript { $_.NextHop -Ne "0.0.0.0" } | Where-Object  
-FilterScript { ($_.NextHop.SubString(0,6) -Ne "fe80::") }
```

This command gets IP routes that have next hops that are not in the local subnet. The command gets all routes, and then passes them to a series of `Where-Object` commands by using the pipeline operator. The command uses different filter scripts to discard routes that are the default gateway for the two IP address families and the IPv6 addresses that begin with `fe80`. For more information about filtering by using `Where-Object`, type ``Get-Help Where-Object``.

Example 6: Get network adapters that have IP routes to non-local destinations

```
PS C:\>Get-NetRoute | Where-Object -FilterScript { $_.NextHop -Ne "::"} |  
Where-Object -FilterScript { $_.NextHop -Ne "0.0.0.0" } | Where-Object  
-FilterScript { ($_.NextHop.SubString(0,6) -Ne "fe80::") } | Get-NetAdapter
```

This command gets network adapters that have IP routes that have next hops that are not in the local subnet. As in the previous example, the command gets the routes that have next hop values by using the `Get-NetRoute` and the `Where-Object` cmdlets, and then passes those routes to the `Get-NetAdapter` cmdlet by using the pipeline operator.

Example 7: Get IP routes that have an infinite valid lifetime

```
PS C:\>Get-NetRoute | Where-Object -FilterScript { $_.ValidLifetime -Eq  
([TimeSpan]::MaxValue) }
```

This command gets all IP routes, and then passes them to the `Where-Object` cmdlet by using the pipeline operator. The command selects those routes that have a valid lifetime of the maximum value.

REMARKS

To see the examples, type: "get-help Get-NetRoute -examples".

For more information, type: "get-help Get-NetRoute -detailed".

For technical information, type: "get-help Get-NetRoute -full".

For online help, type: "get-help Get-NetRoute -online"