# Netsh commands for Routing

Updated: January 21, 2005

## Netsh commands for Routing

You can use the Netsh commands for Routing to administer routing servers from the command line, rather than through the Routing and Remote Access console. Administering servers from the command line is most useful in the following situations:

- You can administer routing servers more quickly from the command line than you can using a console. The ability is especially important if you are using a slow-speed network link, for example, a connection over a wide area network (WAN).
- You can administer routing servers more consistently by incorporating these commands into a script that you can run on many servers.

You can use the following commands at the command prompt for the appropriate Netsh context to administer routing servers. To run these Netsh commands on a remote Windows 2000 Server, you must first use Remote Desktop Connection to connect to a Windows 2000 Server that is running Terminal server. There might be functional differences between Netsh context commands on Windows 2000 and the Windows Server 2003 family.

For more information on Netsh, see Netsh overview [http://technet2.microsoft.com/WindowsServer/en/library/61427fbd-de1f-4c8a-b613-321f7a3cca6a1033.mspx] and Enter a netsh context [http://technet2.microsoft.com/WindowsServer/en/library/d9b4eed7-f79b-4daf-8c22-ffd9428ddea51033.mspx] .

## Netsh Routing IP IGMP commands

The following commands are available for use in the Netsh Routing IP IGMP context.

To view the command syntax, click a command:

- add interface
- install
- delete interface
- uninstall
- set interface
- set global
- show interface
- show global
- show ifstats
- show iftable
- show grouptable
- show rasgrouptable
- show proxygrouptable

### add interface

Configures IGMP on the specified interface.

#### Syntax

**add interface** [**InterfaceName=**][**InterfaceName=**]*InterfaceName* [[**IgmpPrototype=**]{**igmprtrv1** | **igmprtrv2** | **igmprtrv3** | **igmpproxy**}] [[**IfEnabled=**]{**enable** | **disable**}] [[**RobustVar**=]*Integer*] [[**GenQueryInterval=**]*Integer*] [[**GenQueryRespTime=**]*Integer*] [[**StartUpQueryCount=**]*Integer*] [[**StartUpQueryInterval=**]*Integer*] [[**LastMemQueryCount=**]*Integer*] [[**LastMemQueryInterval=***Integer*] [[**AccNonRtrAlertPkts=**]{**yes** | **no**}]

### Parameters

**[ InterfaceName=][ InterfaceName=] *InterfaceName***

Required. Specifies the name of the interface on which you want to configure IGMP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ IgmpPrototype=]{ igmprtrv1| igmprtrv2| igmprtrv3| igmpproxy}**

Specifies the type of protocol to be configured on the interface. The default type is igmprtrv2.

**[ IfEnabled=]{ enable| disable}**

Specifies whether the protocol you are configuring should be enabled or disabled. The default is enabled.

**[ RobustVar=]= *Integer***

Specifies the robustness variable. The value should be greater than zero (0). The default value is 2.

**[ GenQueryInterval=] *Integer***

Specifies the interval, in seconds, at which general queries should be sent on the interface. The default value is 125 seconds.

**[ GenQueryRespTime=] *Integer***

Specifies the maximum response time, in seconds, by which hosts should respond to a general query. The default value is 10 seconds.

**[ StartUpQueryCount=] *Integer***

Specifies the number of general queries that will be sent during startup. The default value is 2.

**[ StartUpQueryInterval=] *Integer***

Specifies the interval, in seconds, between general queries sent during startup. The default value is 31 seconds.

**[ LastMemQueryCount=] *Integer***

Specifies the number of group-specific queries sent when notice of a host leaving the group is received. The default value is 2.

**[ LastMemQueryInterval=] *Integer***

Specifies the interval, in milliseconds, between group-specific queries. The default value is 1000 milliseconds.

**[ AccNonRtrAlertPkts=]{ yes| no}**

Specifies whether to accept IGMP packets that might not have the router alert option set. The default is **yes**.

### Examples

To modify the default startup query interval to 21 seconds for use with IGMP configuration of the interface named Local Area Connection, type:

**add interface "Local Area Connection" startupqueryinterval=21**

Top of page

## install

Installs an IGMP router/proxy and sets global logging. Used without parameters, **install** installs the IGMP router/proxy and specifies that errors related to IGMP are logged but no other IGMP-related events are logged.

### Syntax

**install** [[**LogLevel**=]{**none** | **error** | **warn** | **info**}]

### Parameters

**[ LogLevel=]{ none| error| warn| info}**

Specifies which events should be logged. The **none** parameter specifies no events related to IGMP should be logged. The **error** parameter specifies that only errors related to IGMP should be logged. The **warn** parameter specifies that only warnings related to IGMP should be logged. The **info** parameter specifies that all events related to IGMP should be logged.

Top of page

### delete interface

Removes an IGMP router or proxy from the specified interface.

#### Syntax

**delete interface** [**InterfaceName=**][**InterfaceName=**]*InterfaceName*

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, an interface from which you want to remove an IGMP router or proxy. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

#### Examples

**delete interface "Local Area Connection"**

Top of page

### uninstall

Removes IGMP router/proxy from all interfaces.

#### Syntax

**uninstall**

#### Parameters

none

Top of page

### set interface

Changes interface configuration parameters.

#### Syntax

**set interface** [**InterfaceName=**]*InterfaceName* [[**IgmpPrototype=**]{**igmprtrv1** | **igmprtrv2** | **igmprtrv3** | **igmpproxy**}] [[**IfEnabled=**]{**enable** | **disable**}] [[**RobustVar=**]*Integer*] [[**GenQueryInterval=**]*Integer*] [[**GenQueryRespTime=**]*Integer*] [[**StartUpQueryCount=**]*Integer*] [[**StartUpQueryInterval=**]*Integer*] [[**LastMemQueryCount=**]*Integer*] [[**LastMemQueryInterval=**]*Integer*] [[**AccNonRtrAlertPkts=**]{**yes** | **no**}]

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies the name of the interface on which you want to configure IGMP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ IgmpPrototype=]{ igmprtrv1| igmprtrv2| igmprtrv3| igmpproxy}**

Specifies the type of protocol to be configured on the interface. The default type is igmprtrv2.

**[ IfEnabled=]{ enable| disable}**

Specifies whether the protocol you are configuring should be enabled or disabled. The default is enabled.

**[ RobustVar=] *Integer***

Specifies the robustness variable. The value should be greater than zero (0). The default value is 2.

**[ GenQueryInterval=] *Integer***

Specifies the interval, in seconds, at which general queries should be sent on the interface. The default value is 125

seconds.

**[ GenQueryRespTime=]** ***Integer***

Specifies the maximum response time, in seconds, by which hosts should respond to a general query. The default value is 10 seconds.

**[ StartUpQueryCount=]** ***Integer***

Specifies the number of general queries that will be sent during startup. The default value is 2.

**[ StartUpQueryInterval=]** ***Integer***

Specifies the interval, in seconds, between general queries sent during startup. The default value is 31 seconds.

**[ LastMemQueryCount=]** ***Integer***

Specifies the number of group-specific queries sent when notice of a host leaving the group is received. The default value is 2.

**[ LastMemQueryInterval=]** ***Integer***

Specifies the interval, in milliseconds, between group-specific queries. The default value is 1,000 milliseconds.

**[ AccNonRtrAlertPkts=]{ yes| no}**

Specifies whether to accept IGMP packets that do not have the router alert option set. The default is to accept those packets.

**Examples**

To modify the default startup query interval to 21 seconds for use with IGMP configuration of an interface named Local Area Connection, type:

**set interface "Local Area Connection" startupqueryinterval=21**

Top of page

### set global

Sets IGMP global parameters. Used without parameters, **set global** specifies that warnings are logged but that no other events related to IGMP are logged.

**Syntax**

**set global** [**LogLevel**=]{**none** | **error** | **warn** | **info**}

**Parameters**

**[ LogLevel=]{ none| error| warn| info}**

Required. Specifies which events should be logged. The **none** parameter specifies that no events related to IGMP should be logged. The **error** parameter specifies that only errors related to IGMP should be logged. The **warn** parameter specifies that only warnings related to IGMP should be logged. The **info** parameter specifies that all events related to IGMP should be logged.

Top of page

### show interface

Displays the IGMP configuration of the specified interface. Used without parameters, **show interface** displays IGMP configuration information for all interfaces.

**Syntax**

**show interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface for you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### show global

Displays the global IGMP parameters.

#### Syntax

**show global**

#### Parameters

none

Top of page

### show ifstats

Displays IGMP statistics for the specified interface. Used without parameters, **show ifstats** shows IGMP statistics once for all interfaces.

#### Syntax

**show ifstats** [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

#### Parameters

**[ Index=] *Integer***

Specifies a number identifying the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

#### Examples

To show IGMP statistics every 5 seconds for an interface with an index value of 1001, type:

**show ifstats index=1001 rr=5**

Top of page

### show iftable

Shows IGMP host groups for a specified interface. Used without parameters, **show iftable** shows IGMP host groups for all interfaces once.

#### Syntax

**show iftable** [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

#### Parameters

**[ Index=] *Integer***

Specifies, by index number, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

#### Examples

To show IGMP host groups every 5 seconds for an interface with an index value of 1001, type:

**show iftable index=1001 rr=5**

Top of page

### show grouptable

Shows the IGMP hosts group table for a multicast group. Used without parameters, **show grouptable** shows the IGMP hosts

group tables once for all multicast groups on all interfaces associated with the router.

**Syntax**

**show grouptable** [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

**Parameters**

**[ Index=] *Integer***

Specifies, by IP address, the multicast group for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

**Examples**

To display the IGMP hosts group table every 5 seconds for an IGMP multicast group identified by the IP address of 192.168.100.239, type:

**show grouptable index=192.168.100.239 rr=5**

Top of page

## show rasgrouptable

Displays the hosts group table for a remote access client interface. Used without parameters, **show rasgrouptable** displays the hosts group tables once for all remote access client interfaces.

**Syntax**

**show rasgrouptable** [[**Index**=]*IPAddress*] [[**Rr**=]*Integer*]

**Parameters**

**[ Index=] *IPAddress***

Specifies, by IP address, the remote access client interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

**Examples**

To display the hosts group table for a remote access client interface with an IP address of 10.10.1.100 and to refresh the information every five seconds, type:

**show rasgrouptable index=10.10.1.100 rr=5**

Top of page

## show proxygrouptable

Displays the IGMP hosts group table for an IGMP proxy interface. Used without parameters, **show proxygrouptable** displays information for all interfaces once.

**Syntax**

**show proxygrouptable** [[**InterfaceName=**]*InterfaceName*] [[**Rr**=]*Integer*]

**Parameters**

**[ InterfaceName=] *InterfaceName***

Specifies, by name, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

**Examples**

To display IGMP hosts group tables for all IGMP proxy interfaces and to refresh the information every five seconds, type:

**show proxygrouptable rr=5**

Top of page

## Netsh Routing IP Routerdiscovery commands

The following commands are available for use in the Netsh Routing IP Routerdiscovery context.

To view the command syntax, click a command:

- add interface
- set interface
- delete interface
- show interface

### add interface

Configures router discovery for the specified interface.

#### Syntax

**add interface** [**InterfaceName=**]*InterfaceName* [[**Disc=**]{**enable** | **disable**}] [[**MinInt=**]*Integer*] [[**MaxInt=**]*Integer*] [[**Life=**]*Integer*] [[**Level=**]*Integer*]

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to configure router discovery. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Disc=]{ enable| disable}**

Specifies whether router discovery should occur on this interface. The **enable** parameter specifies that router discovery should occur, and the **disable** parameter specifies that router discovery should not occur.

**[ MinInt=] *Integer***

Specifies a minimum interval, in minutes, between when router discovery packets are sent.

**[ MaxInt=] *Integer***

Specifies a maximum interval, in minutes, between when router discovery packets are sent.

**[ Life=] *Integer***

Specifies the number of minutes for which a router discovery packet sent from this router or server is valid.

**[ Level=] *Integer***

Specifies a number indicating preference for using this interface as the default gateway. Where multiple interfaces are enabled for router discovery, a higher number used here indicates a higher level of preference.

#### Remarks

- When added, router discovery advertisements are sent at random times bounded by the set minimum and maximum intervals.

#### Examples

To enable router discovery for the Local Area Connection interface, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type:

**add interface "Local Area Connection" enable 7 10 30 0**

Top of page

### set interface

Configures router-discovery for an interface.

#### Syntax

**set interface** [**InterfaceName=**]*InterfaceName* [[**Disc=**]{**enable** | **disable**}] [[**MinInt=**]*Integer*] [[**MaxInt=**]*Integer*] [[**Life=**]*Integer*] [[**Level=**]*Integer*]

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to configure router discovery. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Disc=]{ enable| disable}**

Specifies whether router discovery should occur on this interface. The **enable** parameter specifies that router discovery should occur, and the **disable** parameter specifies that router discovery should not occur.

**[ MinInt=] *Integer***

Specifies, in minutes, a minimum interval between when router discovery packets are sent.

**[ MaxInt=] *Integer***

Specifies, in minutes, a maximum interval between when router discovery packets are sent.

**[ Life=] *Integer***

Specifies, in minutes, how long a router discovery packet sent from this router or server is valid.

**[ Level=] *Integer***

Specifies a number indicating preference for using this interface as the default gateway. A higher number used here specifies a higher level of preference.

#### Examples

To enable router discovery for the Local Area Connection interface, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type:

**set interface "Local Area Connection" enable 7 10 30 0**

Top of page

### delete interface

Deletes router-discovery configuration.

#### Syntax

**delete interface** [**InterfaceName=**]*InterfaceName*

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to delete router-discovery configuration. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

#### Examples

To delete the router-discovery configuration for an interface named Local Area Connection, type:

**delete interface name="Local Area Connection"**

Top of page

### show interface

Displays router-discovery information. Used without parameters, **show interface** displays information for all interfaces.

**Syntax**

**show interface** [[**InterfaceName=**]*InterfaceName*]

**Parameters**

**[ InterfaceName=] *InterfaceName***

Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**Examples**

To display router-discovery information for an interface named Local Area Connection, type:

**show interface "Local Area Connection"**

Top of page

## Netsh Routing IP RIP commands

The following commands are available for use in the Netsh Routing IP RIP context.

To view the command syntax, click a command:



### add peerfilter

Adds a filter for servers that can be accepted as peers.

**Syntax**

**add peerfilter** [**Server**=]*IPAddress*

**Parameters**

**[ Server=]** ***IPAddress***

Required. Specifies, by IP address, the peer server whose traffic you want to filter.

Top of page

### add acceptfilter

Adds an acceptance filter for routes received on an interface.

#### Syntax

**add acceptfilter** [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1IPAddress2*

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Addr=]** ***IPAddress1IPAddress2***

Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address to use in building the filter.

#### Examples

To add an acceptance filter for an interface named Local Area Connection for an IP range of 10.0.0.2 to 10.11.21.154, type:

**add acceptfilter "Local Area Connection" 10.0.0.2 10.11.21.154**

Top of page

### add announcefilter

Adds a filter for routes announced on an interface.

#### Syntax

**add announcefilter** [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1IPAddress2*

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies the name of an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Addr=]** ***IPAddress1IPAddress2***

Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address to use in building the filter.

#### Examples

To add a filter for routes announced on an interface named Local Area Connection for a range of IP addresses from 10.0.0.2 to 10.11.21.154, type:

**add announcefilter Local Area Connection 10.0.0.2 10.11.21.154**

Top of page

### add neighbor

Adds a RIP neighbor on an interface.

#### Syntax

**add neighbor** [**InterfaceName=**]*InterfaceName* [**addr=**]*IPAddress*

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ addr=] *IPAddress***

Required. Specifies the IP address of the RIP neighbor you want to add.

### Examples

To add a RIP neighbor located at an IP address of 10.0.0.2 for an interface named Local Area Connection, type:

**add neighbor "Local Area Connection" 10.0.0.2**

Top of page

## add interface

Adds RIP to a specified interface.

### Syntax

**add interface** [**InterfaceName=**]*InterfaceName* [[**Metric=**]*Integer*] [[**UpdateMode=**]{**demand** | **periodic**}] [[**Announce=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Accept=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Expire=**]*Integer*] [[**Remove=**]*Integer*] [[**Update=**]*Integer*] [[**AuthMode=**]{**authnone** | **authsimplepassword**}] [[**Tag=**]*Integer*] [[**Unicast=**]{**also** | **only** | **disable**}] [[**AccFiltMode=**]{**include** | **exclude** | **disable**}] [[**AnnFiltMode=**]{**include** | **exclude** | **disable**}] [[**Password=**]*String*]

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which to add RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Metric=] *Integer***

Specifies a metric value for routes based on this interface.

**[ UpdateMode=]{ demand| periodic}**

Specifies when updates will occur. The **demand** parameter specifies that updates should occur on demand. The **periodic** parameter specifies that updates should occur at regular intervals.

**[ Announce=]{ none| rip1| rip1compat| rip2}**

Specifies which routes should be announced. The **none** parameter specifies that all routes should be announced. The **rip1** parameter specifies that only RIP version 1 routes should be announced. The **rip1compat** parameter specifies that announcements should be made in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be announced.

**[ Accept=]{ none| rip1| rip1compat| rip2}**

Specifies which routes should be accepted. The **none** parameter specifies that all routes should be accepted. The **rip1** parameter specifies that only RIP version 1 routes should be accepted. The **rip1compat** parameter specifies that routes should be accepted in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be accepted.

**[ Expire=] *Integer***

Specifies, in seconds, the route expiration interval.

**[ Remove=] *Integer***

Specifies, in seconds, the route removal interval.

**[ Update=] *Integer***

Specifies, in seconds, the full update interval.

**[ AuthMode=]{ authnone| authsimplepassword}**

Specifies whether authorization should be required. The **authnone** parameter specifies that no authorization should be

required. The **authsimplepassword** parameter specifies that password authorization should be required.

**[ Tag=]** ***Integer***

Specifies the route tag.

**[ Unicast=]{ also| only| disable}**

Specifies the unicast mode. The **also** parameter specifies use also unicast mode. The **only** parameter specifies use only unicast mode. The **disable** parameter specifies that unicast mode should be disabled.

**[ AccFiltMode=]{ include| exclude| disable}**

Specifies whether acceptance filters should be included or disabled. The **include** parameter specifies that acceptance filters should be included. The **exclude** parameter specifies that acceptance filters should not be included. The **disable** parameter specifies that acceptance filters should be disabled.

**[ AnnFiltMode=]{ include| exclude| disable}**

Specifies whether announcement filters should be included or disabled. The **include** parameter specifies that announcement filters should be included. The **exclude** parameter specifies that announcement filters should not be included. The **disable** parameter specifies that announcement filters should be disabled.

**[ Password=]** ***String***

Specifies a plaintext password no longer than 16 characters.

Top of page

### add helper

Installs a helper DLL under RIP.

#### Syntax

**add helper***DLLName*

#### Parameters

***DLLName***

Required. Specifies a DLL that supports this context.

Top of page

### delete peerfilter

Deletes a filter for an accepted peer server.

#### Syntax

**delete peerfilter** [**Server**=]*IPAddress*

#### Parameters

**[ Server=]** ***IPAddress***

Required. Specifies, by IP address, the peer server whose traffic you do not want to filter.

Top of page

### delete acceptfilter

Deletes an acceptance filter for routes received on a specified interface.

#### Syntax

**delete acceptfilter** [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1IPAddress2*

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the

name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

**[ Addr=] *IPAddress1IPAddress2***

Required. Specifies an IP address pairing. The first address is for the low IP address, and the second address is for the high IP address used in building the filter.

### Examples

To delete an acceptance filter for the IP ranges of 10.0.0.2 to 10.11.21.154 on an interface named Local Area Connection, type:

**delete acceptfilter "Local Area Connection" 10.0.0.2 10.11.21.154**

Top of page

## delete announcefilter

Deletes an announcement filter set for an interface.

### Syntax

**delete announcefilter** [**InterfaceName=**]*InterfaceName* [**Addr=**]*IPAddress1IPAddress2*

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies the name of an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

**[ Addr=] *IPAddress1IPAddress2***

Required. Specifies an IP address pairing. The first address specifies the low IP address, and the second address specifies the high IP address used in building the filter.

### Examples

To delete an announcement filter for the IP range 10.0.0.2 to 10.11.21.154 for an interface named Local Area Connection, type:

**delete announcefilter "Local Area Connection" 10.0.0.2 10.11.21.154**

Top of page

## delete neighbor

Deletes a RIP neighbor from an interface.

### Syntax

**delete neighbor** [**InterfaceName=**]*InterfaceNameIPAddress*

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, an interface where RIP is available for use. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

***IPAddress***

Required. Specifies, by IP address, the RIP neighbor to delete.

### Examples

To delete a RIP neighbor located at 10.0.0.2 for an interface named Local Area Connection, type:

**delete neighbor "Local Area Connection" 10.0.0.2**

Top of page

### delete interface

Removes RIP from the specified interface.

#### Syntax

**delete interface** [**InterfaceName=**]*InterfaceName*

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, an interface from which you want to remove RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### delete helper

Removes a specified helper DLL from Netsh.

#### Syntax

**delete helper***HelperName*

#### Parameters

***HelperName***

Required. Specifies, by file name, the helper DLL to remove.

Top of page

### set interface

Configures RIP on a specified interface.

#### Syntax

**set interface** [**InterfaceName=**]*InterfaceName* [[**Metric=**]*Integer*] [[**UpdateMode=**]{**demand** | **periodic**}] [[**Announce=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Accept=**]{**none** | **rip1** | **rip1compat** | **rip2**}] [[**Expire=**]*Integer*] [[**Remove=**]*Integer*] [[**Update=**]*Integer*] [[**AuthMode=**]{**authnone** | **authsimplepassword**}] [[**Tag=**]*Integer*] [[**Unicast=**]{**also** | **only** | **disable**}] [[**AccFiltMode=**]{**include** | **exclude** | **disable**}] [[**AnnFiltMode=**]{**include** | **exclude** | **disable**}] [[**Password=**]*String*]

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which to configure RIP. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ Metric=] *Integer***

Specifies a metric value for routes based on this interface.

**[ UpdateMode=]{ demand| periodic}**

Specifies when updates should occur. The **demand** parameter specifies that updates should occur on demand. The **periodic** parameter specifies that updates should occur at regular intervals.

**[ Announce=]{ none| rip1| rip1compat| rip2}**

Specifies which routes should be announced. The **none** parameter specifies that all routes should be announced. The **rip1** parameter specifies that only RIP version 1 routes should be announced. The **rip1compat** parameter specifies that announcements should be made in RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be announced.

**[ Accept=]{ none| rip1| rip1compat| rip2}**

Specifies which routes should be accepted. The **none** parameter specifies that all routes should be accepted. The **rip1** parameter specifies that only RIP version 1 routes should be accepted. The **rip1compat** parameter specifies that acceptance should occur using RIP version-compatible mode. The **rip2** parameter specifies that only RIP version 2 routes should be accepted.

**[ Expire=]** ***Integer***

Specifies, in seconds, how much time should elapse before the route expires.

**[ Remove=]** ***Integer***

Specifies, in seconds, how much time should elapse before the route is removed.

**[ Update=]** ***Integer***

Specifies, in seconds, how often full updates occur.

**[ AuthMode=]{ authnone| authsimplepassword}**

Specifies whether authorization should be required. The **authnone** parameter specifies that no authorization should be required. The **authsimplepassword** parameter specifies that password authorization should be required.

**[ Tag=]** ***Integer***

Specifies the route tag.

**[ Unicast=]{ also| only| disable}**

Specifies the unicast mode. The **also** parameter specifies use also unicast mode. The **only** parameter specifies use only unicast mode. The **disable** parameter specifies that unicast mode should be disabled.

**[ AccFiltMode=]{ include| exclude| disable}**

Specifies whether acceptance filters should be included. The **include** parameter specifies that acceptance filters should be included. The **exclude** parameter specifies that acceptance filters should not be included. The **disable** parameter specifies that acceptance filters should be disabled.

**[ AnnFiltMode=]{ include| exclude| disable}**

Specifies whether announcement filters should be included or disabled. The **include** parameter specifies that announcement filters should be included. The **exclude** parameter specifies that announcement filters should not be included. The **disable** parameter specifies that announcement filters should be disabled.

**[ Password=]** ***String***

Specifies a plaintext password no longer than 16 characters long.

**Examples**

To modify the RIP configuration on the interface named Local Area Connection, specifying a metric value of 2 and that updates should occur on demand, type:

**set interface "Local Area Connection" metric=2 updatemode=periodic**

Top of page

## set flags

Sets RIP-related flags for a specified interface.

**Syntax**

**set flags** [**InterfaceName=**]*InterfaceName* [**clear,**][**splithorizon,**][**poisonreverse,**][**triggeredupdates,**] [**cleanupupdates**][**accepthostroutes,**][**acceptdefaultroutes,**][**senddefaultroutes,**][**nosubnetsummary**]

**Parameters**

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface on which to set flags. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, "*Interface Name*").

**[ clear,][ splithorizon,][ poisonreverse,][ triggeredupdates,][ cleanupupdates,][ accepthostroutes,] [ acceptdefaultroutes,][ senddefaultroutes,][ nosubnetsummary]**

Specifies policies for RIP-related flags. The **clear** parameter specifies that all set RIP flags should be cleared. The **splithorizon** parameter specifies that RIP for split horizon should be enabled. The **poisonreverse** parameter specifies that RIP for poison reverse should be enabled. The **triggeredupdates** parameter specifies that RIP triggered updates should occur. The **cleanupupdates** parameter specifies that cleanup updates for RIP should occur. The **accepthostroutes** parameter specifies that host routes should be accepted. The **acceptdefaultroutes** parameter specifies that default routes should be accepted. The **senddefaultroutes** parameter specifies that default routes should be sent. The **nosubnetsummary** parameter specifies that subnet summarization should not occur.

**Remarks**

-

The last policy parameter should not have a comma after it.

Top of page

### set global

Sets global RIP parameters.

#### Syntax

**set global** {[[**loglevel**=]{**none** | **error** | **warn** | **info**}] | [[**mintrig**=]*Integer*] | [[**peermode**=]{**include** | **exclude** | **disable**}]}

#### Parameters

**[ loglevel=]{ none| error| warn| info}**

Specifies which RIP events should be logged. The **none** parameter specifies that no RIP events should be logged. The **error** parameter specifies that only errors related to RIP should be logged. The **warn** parameter specifies that only warnings related to RIP should be logged. The **info** parameter specifies that all events related to RIP should be logged.

**[ mintrig=]** ***Integer***

Specifies, in minutes, the minimum amount of time that should elapse between triggers.

**[ peermode=]{ include| exclude| disable}**

Specifies whether peers should be included and whether peer mode should be disabled. The **include** parameter specifies that peers should be included.The **exclude** parameter specifies that peers should not be included.The **disable** parameter specifies that peer mode should be disabled.

Top of page

### show interface

Shows RIP configuration for the specified interface. Used without parameters, **show interface** displays configuration information for all interfaces.

#### Syntax

**show interface** [[**InterfaceName=**]*InterfaceName*]

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Specifies, by name, the interface whose information you want to display. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### show flags

Displays RIP flags set for a specified interface. Used without parameters, **show flags** displays information for all interfaces.

#### Syntax

**show flags** [[**InterfaceName=**]*InterfaceName*]

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### show global

Shows RIP global parameters.

#### Syntax

**show global**

#### Parameters

none

Top of page

### show ifstats

Displays RIP statistics for a specified interface. Used without parameters, **show ifstats** displays information once for all interfaces.

#### Syntax

**show ifstats** [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

#### Parameters

**[ Index=] *Integer***

Specifies, by index number, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

#### Remarks

- To list the index value for each interface, use the **show interface** command in the Netsh Interface IP context.

#### Examples

To display RIP statistics every 5 seconds for an interface with an index value of 1, type:

**show ifstats index=1 rr=5**

Top of page

### show ifbinding

Displays RIP interface IP address binding. Used without parameters, **show ifbinding** displays information once for all interfaces.

#### Syntax

**show ifbinding** [[**Index**=]*Integer*] [[**Rr**=]*Integer*]

#### Parameters

**[ Index=] *Integer***

Specifies, by index number, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

#### Remarks

- To list the index value for each interface, use the **show interface** command in the Netsh Interface IP context.

#### Examples

To display the bindings every 5 seconds for an interface with an index value of 1, type:

**show ifbinding index=1 rr=5**

Top of page

## Netsh Routing IP DHCP Relay Agent commands

The following commands are available for use in the Netsh Routing IP DHCP Relay Agent context.

To view the command syntax, click a command:



### add dhcpserver

Adds a DHCP server to the list of DHCP servers whose messages will be forwarded by DHCP Relay Agent.

**Syntax**

**add dhcpserver***IPAddress*

**Parameters**

***IPAddress***

Required. Specifies, by IP address, the DHCP server you want to add.

Top of page

### add interface

Enables DHCP Relay Agent on the specified interface.

**Syntax**

**add interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to enable DHCP Relay Agent.

Top of page

### delete interface

Disables DHCP Relay Agent on the specified interface.

**Syntax**

**delete interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to disable DHCP Relay Agent. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

## delete dhcpserver

Removes a DHCP server from the list of DHCP servers whose messages will be forwarded by DHCP Relay Agent.

**Syntax**

**delete dhcpserver***IPAddress*

**Parameters**

***IPAddress***

Required. Specifies, by IP address, the DHCP server to be deleted.

Top of page

## set global

Sets global parameters for DHCP Relay Agent configuration.

**Syntax**

**set global** [**LogLevel=**]{**none** | **error** | **warn** | **info**}

**Parameters**

**[ LogLevel=]{ none| error| warn| info}**

Required. Specifies which DHCP Relay Agent events should be logged. The **none** parameter specifies that no events related to DHCP Relay Agent should be logged. The **error** parameter specifies that only errors related to DHCP Relay Agent should be logged. The **warn** parameter specifies that only warnings related to DHCP Relay Agent should be logged. The **info** parameter specifies that all events related to DHCP Relay Agent should be logged.

Top of page

## set interface

Updates the DHCP Relay Agent configuration on an interface.

**Syntax**

**set interface** [**InterfaceName=**]*InterfaceName* [**relaymode=**]{**enable** | **disable**} [[**maxhop=**]*Integer*] [[**minsecs=**] *Integer*]

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to update the configuration of DHCP Relay Agent. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ relaymode=]{ enable| disable}**

Required. Specifies whether the DHCP Relay Agent is enabled or disabled. The **enable** parameter specifies that DHCP Relay Agent should be enabled for this interface. The **disable** parameter disables DHCP Relay Agent for this interface.

**[ maxhop=] *Integer***

Specifies the number of hops a DHCP packet can make before it should be dropped and should no longer be relayed.

**[ minsecs=] *Integer***

Specifies the minimum number of seconds from boot time that must appear in a DHCP packet before it should be sent to a DHCP server in the list of servers whose messages will be forwarded by DHCP Relay Agent.

**Examples**

To update the DHCP Relay Agent configuration on the interface named Local Area Connection with a maximum hop count of 16 and a value of 4 seconds for the minimum seconds from boot time, type:

**set interface Local Area Connection enable 16 4**

Top of page

### show interface

Displays configuration information for DHCP Relay Agent on the specified interface.

**Syntax**

**show interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### show global

Displays DHCP Relay Agent global configuration.

**Syntax**

**show global** [[**Rr=**]*Integer*]

**Parameters**

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

Top of page

### show ifconfig

Displays DHCP Relay Agent configuration information for the specified interface. Used without parameters, **show ifconfig** displays information once for all interfaces where DHCP Relay Agent is enabled.

**Syntax**

**show ifconfig** [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

**Parameters**

**[ Index=] *Integer***

Specifies, by index value, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

Top of page

### show ifbinding

Displays IP address bindings for the specified interface. Used without parameters, **show ifbinding** displays information once for all interfaces where DHCP relay is enabled for use.

**Syntax**

**show ifbinding** [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

**Parameters**

**[ Index=] *Integer***

Specifies, by index value, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

Top of page

### show ifstats

Displays DHCP Relay Agent statistics for the specified interface. Used without parameters, **show ifstats** displays information once for all interfaces where DHCP relay is enabled for use.

**Syntax**

**show ifstats** [[**Index**=]*Integer*] [[**Rr=**]*Integer*]

**Parameters**

**[ Index=] *Integer***

Specifies, by index value, the interface for which you want to display information.

**[ Rr=] *Integer***

Specifies, in seconds, how often you want the information to be refreshed.

Top of page

## Netsh Routing IP OSPF commands

The following commands are available for use in the Netsh Routing IP OSPF context.

**Notes**

- This feature is not available on the Itanium-based versions of the Windows operating systems.
- This content is not available in this preliminary release.x64

To view the command syntax, click a command:

- delete neighbor
- delete interface
- delete routefilter
- delete protofilter
- uninstall
- set area
- set virtif
- set interface
- set global
- set routefilter
- set protofilter
- show global
- show area
- show virtif
- show interface
- show routefilter
- show protofilter
- show area
- show virtif

### add range

Adds the IP address and mask pair that define a range of addresses belonging to this area.

#### Syntax

**add range** [**Areaid=**]*IPAddress* [**Range=**]*IPAddressSubnetMask*

#### Parameters

**[ Areaid=] *IPAddress***

Required. Specifies the IP address of the area to which the range is added.

**[ Range=] *IPAddressSubnetMask***

Required. Specifies the IP address and the mask of the range to add.

#### Remarks

- OSPF ranges are used to summarize the routes within the OSPF area.

Top of page

### add area

Adds an ID for the area.

#### Syntax

**add area** [**Areaid=**]*IPAddress*

#### Parameters

**[ Areaid=] *IPAddress***

Required. Specifies the IP address of the area to which you want to add the range.

**Remarks**

- The **Areaid** parameter is a 32-bit number expressed in dotted decimal notation that identifies the OSPF area. The area ID does not need to coincide with an IP address or an IP network ID. The area ID of 0.0.0.0 is reserved for the backbone. If the area represents a subnetted network, you can use the IP network number of the subnetted network for the area ID.

Top of page

### add virtif

Adds the virtual interface for a router.

**Syntax**

**add virtif** [**transareaid=**]*TransitAreaID* [**virtnbrid=**]*VirtualNeighborID*

**Parameters**

**[ transareaid=] *TransitAreaID***

Required. Specifies the IP address of the transit area used to exchange routing information between the backbone area border router and the area border router of the noncontiguous area.

**[ virtnbrid=] *VirtualNeighborID***

Required. Specifies, by IP address, the router ID of the virtual neighbor to add.

Top of page

### add neighbor

Adds an OSPF neighbor to the specified interface.

**Syntax**

**add neighbor** [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**nbraddr**=]*IPAddress* [**nbrprio=**]*Priority*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface to which you want to add a neighbor. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ addr=] *IPAddress***

Required. Specifies the OSPF interface, by IP address.

**[ nbraddr=] *IPAddress***

Required. Specifies the IP address of the neighbor you want to add.

**[ nbrprio=] *Priority***

Required. Specifies the OSPF router priority of the neighbor you want to add.

Top of page

### add interface

Configures OSPF on the specified interface.

**Syntax**

**add interface** [**InterfaceName=**]*InterfaceName* [**areaid**=]*IPAddress* [**addr**=]*IPAddress* [**mask**=]*SubnetMask*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to add OSPF. The *InterfaceName* parameter must match the

name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ areaid=]** ***IPAddress***

Required. Specifies the IP address of the area to which the interface belongs.

**[ addr=]** ***IPAddress***

Required. Specifies the IP address of the interface to add.

**[ mask=]** ***SubnetMask***

Required. Specifies the subnet mask of the interface to add.

Top of page

### add routefilter

Adds an OSPF route filter.

#### Syntax

**add routefilter** [**Filter=**]*IPAddressSubnetMask*

#### Parameters

**[ Filter=]** ***IPAddressSubnetMask***

Required. Specifies, by IP address and subnet mask, the route you want to filter.

Top of page

### add protofilter

Adds an OSPF protocol filter.

#### Syntax

**add protofilter** [**filter=**]{**autostatic** | **local** | **rip** | **snmp** | **nondod** | **static**}

#### Parameters

**[ filter=]{ autostatic| local| rip| snmp| nondod| static}**

Required. Specifies the protocol filter you want to add. The **autostatic** parameter specifies a RIP autostatic route. The **local** parameter specifies a route added by the network to which the computer is connected. The **rip** parameter specifies that the filter is added using RIP. The **snmp** parameter specifies that the route is added using SNMP. The **nondodm** parameter specifies an RTM route that is present only when a DOD interface is connected. The **static** parameter specifies an RTM route.

#### Remarks

- The **add protofilter** command is used on an OSPF router acting as an Autonomous System Boundary Router. The filters you add filter other route sources by protocol source, such as static routes and RIP, so that they are not redistributed into the OSPF Autonomous System.

Top of page

### install

Installs OSPF.

#### Syntax

**install**

#### Parameters

none

Top of page

### delete range

Deletes the specified range for the area ID.

#### Syntax

**delete range** [**Areaid=**]*IPAddress* [**Range=**]*IPAddressSubnetMask*

#### Parameters

**[ Areaid=] *IPAddress***

Required. Specifies, by IP address, the area you want to delete.

**[ Range=] *IPAddressSubnetMask***

Required. Specifies the IP address and the subnet mask of the area range you want to delete.

Top of page

### delete area

Deletes the specified area.

#### Syntax

**delete area** [**Areaid=**]*IPAddress*

#### Parameters

**[ Areaid=] *IPAddress***

Required. Specifies, by IP address, the area you want to delete.

#### Remarks

The area ID is a 32-bit number expressed in dotted decimal notation that identifies the OSPF area. The area ID does not need to coincide with an IP address or an IP network ID. The area ID of 0.0.0.0 is reserved for the backbone. If the area represents a subnetted network, you can use the IP network number of the subnetted network for the area ID.

Top of page

### delete virtif

Deletes the specified virtual interface.

#### Syntax

**delete virtif** [**transareaid=**]*TransitAreaID* [**virtnbrid=**]*VirtualNeighborID*

#### Parameters

**[ transareaid=] *TransitAreaID***

Required. Specifies the IP address of the transit area used to exchange routing information between the backbone area border router and the area border router of the noncontiguous area.

**[ virtnbrid=] *VirtualNeighborID***

Required. Specifies the router ID of the virtual neighbor to delete.

Top of page

### delete neighbor

Deletes the specified OSPF neighbor.

#### Syntax

**delete neighbor** [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**nbraddr**=]*IPAddress* [**nbrprio=**]*Priority*

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface from which you want to delete a neighbor. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ addr=]** ***IPAddress***

Required. Specifies, by IP address, the interface from which you want to delete a neighbor.

**[ nbraddr=]** ***IPAddress***

Required. Specifies the IP address of the neighbor to delete.

**[ nbrprio=]** ***Priority***

Required. Specifies the OSPF router priority of the neighbor to delete.

Top of page

### delete interface

Deletes OSPF from the specified interface.

#### Syntax

**delete interface** [**InterfaceName=**]*InterfaceName* [**addr**=]*IPAddress* [**mask**=]*SubnetMask*

#### Parameters

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface from which you want to delete OSPF. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ addr=]** ***IPAddress***

Required. Specifies, by IP address, the interface to delete.

**[ mask=]** ***SubnetMask***

Required. Specifies, by subnet mask, the interface to delete.

Top of page

### delete routefilter

Deletes the specified OSPF route filter.

#### Syntax

**delete routefilter** [**Filter=**]*IPAddressSubnetMask*

#### Parameters

**[ Filter=]** ***IPAddressSubnetMask***

Required. Specifies the IP address and the subnet mask of the route filter you want to delete.

Top of page

### delete protofilter

Deletes the specified OSPF protocol filter.

#### Syntax

**delete protofilter** [**filter=**]{**autostatic** | **local** | **rip** | **snmp** | **nondod** | **static**}

#### Parameters

**[ filter=]{ autostatic| local| rip| snmp| nondod| static}**

Required. Specifies the protocol filter you want to delete. The **autostatic** parameter specifies a RIP autostatic route. The **local** parameter specifies a route added by the network to which the computer is connected. The **rip** parameter specifies that the filter is added using RIP. The **snmp** parameter specifies that the route is added using SNMP. The **nondodm** parameter specifies an RTM route that is present only when a DOD interface is connected. The **static** parameter specifies an RTM route.

**Remarks**

- The **delete protofilter** command is used on an OSPF router acting as an Autonomous System Boundary Router. The filters you add filter other route sources by protocol source, such as static routes and the Routing Information Protocol (RIP), so that they are not redistributed into the OSPF Autonomous System.

Top of page

### uninstall

Deletes OSPF.

**Parameters**

none

Top of page

### set area

Changes parameters for the specified area.

**Syntax**

**set area** [**areaid=**]*AreaID* [[**auth=**]{**none** | **password**}] [[**stubarea**=]{**yes** | **no**}] [[**metric**=]*StubMetric*] [[**sumadv**=]{**yes** | **no**}]

**Parameters**

**[ areaid=] *AreaID***

Required. Specifies, by IP address, the OSPF area for which you want to change parameters.

**[ auth=]{ none| password}**

Specifies whether passwords are required for the area. If passwords are required for an area, all interfaces that are in the same area and on the same network segment must use the same password, and interfaces that are in the same area but on different network segments can have different passwords. By default, passwords are required and the password is 12345678. Passwords are transmitted in plaintext format, so this option is for identification, not security.

**[ stubarea=]{ yes| no}**

Specifies whether this area is configured as a stub area, which is an OSPF area that does not enumerate external routes. Routes from outside the OSPF Autonomous System (AS) are not flooded into or through stub areas. Routing to AS external destinations in these areas is based only on a summary default route. This reduces the overhead for the routers in the stub area.

**[ metric=] *StubMetric***

Specifies the cost of the summary default route that the router advertises into the stub area.

**[ sumadv=]{ yes| no}**

Specifies whether to import summary advertisements.

Top of page

### set virtif

Configures parameters for the specified virtual interface.

**Syntax**

**set virtif** [**transareaid=**]*TransitAreaID* [**virtnbrid =**] *VirtualNeighborID* [[**transdelay=**]*TransitDelay*] [[**retrans=**]*ReTransmitInterval*] [[**hello=**]*HelloInterval*] [[**dead=**]*DeadInterval*] [[**password=**]*Password*]

**Parameters**

**[ transareaid=]** ***TransitAreaID***

Required. Specifies the IP address of the transit area.

**[ virtnbrid=]** ***VirtualNeighborID***

Required. Specifies the router ID of the virtual neighbor.

**[ transdelay=]** ***TransitDelay***

Specifies the estimated number of seconds it takes to transmit a link state update packet over this interface. If you do not specify this parameter, the estimated delay is set to 1 second.

**[ retrans=]** ***ReTransmitInterval***

Specifies the number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. A typical value for a local area network is 5 seconds.

**[ hello=]** ***HelloInterval***

Specifies, in seconds, the interval between transmissions of hello packets by the router on the interface. This setting must be the same for all routers that are connected to the same network. A typical value for a local area network is 10 seconds.

**[ dead=]** ***DeadInterval***

Specifies, in seconds, how long a router can fail to respond before a neighboring router should consider this router is malfunctioning. This setting should be an integral multiple of the hello interval (commonly 4). This value must be the same for all OSPF router interfaces that are connected to the same network segment.

**[ password=]** ***Password***

Specifies the password for this interface, if passwords are required for the area. All interfaces that are in the same area and on the same network must use the same password. By default, passwords are required and the password is 12345678.

Top of page

## set interface

Configures OSPF parameters for an interface.

**Syntax**

**set interface** [**interfacename=**]*InterfaceName* [**addr=**]*IPAddress* [**mask=**]*SubnetMask* [[**state=**]{**enable** | **disable**}] [[**areaid=**]*AreaID*] [[**type=**]{**nbma** | **p2p** | **broadcast**}] [[**prio=**]*Priority*] [[**transdelay=**]*TransitDelay*] [[**retrans=**] *ReTransmitInterval*] [[**hello=**]*HelloInterval*] [[**dead=**]*DeadInterval*] [[**poll=**]*PollInterval*] [[**metric=**]*Metric*] [[**password=**] *Password*] [[**mtu=**]*MTUSize*]

**Parameters**

**[ interfacename=]** ***InterfaceName***

Required. Specifies, by name, the interface for which you want to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ addr=]** ***IPAddress***

Required. Specifies the IP address of the interface.

**[ mask=]** ***SubnetMask***

Required. Specifies the subnet mask of the interface.

**[ state=]{ enable| disable}**

Specifies whether the interface is enabled. The **enable** parameter enables the interface. The **disable** parameter disables the interface.

**[ areaid=]** ***AreaID***

Specifies the IP address of the interface area.

**[ type=]{ nbma| p2p| broadcast}**

Specifies the type of OSPF interface for which you want to configure parameters. The **nbma** parameter specifies a non-broadcast multiple access (NBMA) OSPF interface. The **p2p** parameter specifies a point-to-point OSPF interface. The **broadcast** parameter specifies a broadcast OSPF interface.

**[ prio=]** ***Priority***

Specifies the interface priority.

**[ transdelay=]** ***TransitDelay***

Specifies the estimated number of seconds it takes to transmit a link state update packet over this interface. If you do not specify this parameter, the estimated delay is set to 1 second.

**[ retrans=]** ***ReTransmitInterval***

Specifies the number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. A typical value for a local area network is 5 seconds.

**[ hello=]** ***HelloInterval***

Specifies, in seconds, the interval between transmissions of hello packets by the router on the interface. This setting must be the same for all routers that are connected to the same network. A typical value for a local area network is 10 seconds.

**[ dead=]** ***DeadInterval***

Specifies, in seconds, how long a router can fail to respond before a neighboring router should consider this router to be malfunctional. This setting should be an integral multiple of the hello interval (commonly 4). This value must be the same for all OSPF router interfaces connected to the same network segment.

**[ poll=]** ***PollInterval***

Specifies the number of seconds between OSPF network polls for non-broadcast multiple access (NBMA) interfaces only. You should set the poll interval to be at least twice as long as the dead interval. A typical value for an X.25 network is two minutes.

**[ metric=]** ***Metric***

Specifies the cost of sending a packet out of this interface. The value set here is advertised as the link cost for this interface in the link state advertisement of the router. Faster interfaces usually have lower costs. The maximum setting is 32,767.

**[ password=]** ***Password***

Specifies the password for this interface, if passwords are required for the area. All interfaces that are in the same area and on the same network must use the same password. By default, passwords are required and the password is 12345678.

**[ mtu=]** ***MTUSize***

Specifies, in bytes, the maximum size of IP packets carrying OSPF information that can be sent without fragmentation. The default IP MTU for an Ethernet network is 1,500 bytes.

Top of page

## set global

Configures global OSPF parameters.

### Syntax

**set global** [**routerid=**]*IPAddress* [**asborder=**]{**yes** | **no**} [[**loglevel=**]{**none** | **error** | **warn** | **info**}]

### Parameters

**[ routerid=]** ***IPAddress***

Required. Specifies, by IP address, the router for which you want to configure parameters.

**[ asborder=]{ yes| no}**

Required. Specifies whether the router acts as an Autonomous System (AS) boundary router.

**[ loglevel=]{ none| error| warn| info}**

Specifies which events should be logged. The **none** parameter specifies that no events related to OSPF should be logged. The **error** parameter specifies that only errors related to OSPF should be logged. The **warn** parameter specifies that only warnings related to OSPF should be logged. The **info** parameter specifies that all events related to OSPF should be logged.

Top of page

## set routefilter

Configures the OSPF route filter action.

### Syntax

**set routefilter** [**action=**]{**drop** | **accept**}

### Parameters

**[ action=]{ drop| accept}**

Required. Specifies whether the route filter drops or accepts packets. The **drop** parameter specifies that packets should be dropped. The **accept** parameter specifies that packets should be accepted.

Top of page

### set protofilter

Configures OSPF protocol filter action.

#### Syntax

**set protofilter** [**action=**]{**drop** | **accept**}

#### Parameters

**[ action=]{ drop| accept}**

Required. Specifies whether the protocol filter accepts or drops packets. The **drop** parameter specifies that packets should be dropped. The **accept** parameter specifies that packets should be accepted.

Top of page

### show global

Displays global OSPF parameters.

#### Syntax

**show global**

#### Parameters

none

Top of page

### show area

Displays area parameters.

#### Syntax

**show area**

#### Parameters

none

Top of page

### show virtif

Displays parameters for all virtual interfaces.

#### Syntax

**show virtif**

#### Parameters

none

Top of page

### show interface

Displays OSPF configuration information for the specified interface.

#### Syntax

**show interface** [**InterfaceName=**]*InterfaceName*

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### show routefilter

Displays information about the OSPF route filter.

#### Syntax

**show routefilter**

#### Parameters

None

Top of page

### show protofilter

Displays information about the OSPF protocol filter.

#### Syntax

**show protofilter**

#### Parameters

none

Top of page

### show area

Displays information about OSPF areas.

#### Syntax

**show area**

#### Parameters

none

Top of page

### show virtif

Displays information about OSPF virtual interfaces.

#### Syntax

**show virtif**

**Parameters**

none

Top of page

## Netsh Routing IP Autodhcp commands

The following commands are available for use in the Netsh Routing IP Autodhcp context.

To view the command syntax, click a command:



### add exclusion

Adds an exclusion to the DHCP allocator scope.

**Syntax**

**add exclusion***IPAddress*

**Parameters**

***IPAddress***

Required. Specifies an IP address to exclude from the DHCP allocator scope.

Top of page

### delete exclusion

Deletes an exclusion from the DHCP allocator scope.

**Syntax**

**delete exclusion***IPAddress*

**Parameters**

***IPAddress***

Required. Specifies an IP address already excluded from the DHCP allocator scope.

Top of page

### install

Installs the routing protocol under IP.

**Syntax**

**install**

**Parameters**

none

Top of page

### set global

Configures global parameters for DHCP allocation.

**Syntax**

**set global** {[[**scopenetwork=**]*IPAddress*] | [[**scopemask=**]*SubnetMask*] | [[**leasetime=**]*Minutes*] | [[**loglevel=**]{**none** | **error** | **warn** | **info**}]}

**Parameters**

**[ scopenetwork=] *IPAddress***

Specifies the IP address for the DHCP allocator scope.

**[ scopemask=] *SubnetMask***

Specifies the subnet mask associated with the IP address for the scope.

**[ leasetime=] *Minutes***

Specifies, in minutes, the lease duration time.

**[ LogLevel=]{ none| error| warn| info}**

Specifies which events should be logged. The **none** parameter specifies that no events related to DHCP should be logged. The **error** parameter specifies that only errors related to DHCP should be logged. The **warn** parameter specifies that only warnings related to DHCP should be logged. The **info** parameter specifies that all events related to DHCP should be logged.

**Examples**

To set the IP network for DHCP allocation to 10.10.10.0, with an associated scope subnet mask of 255.255.255.0, a lease time of 11520 minutes (8 days), and logging of information for all related events, type:

**set global 10.10.10.0 255.255.255.0 11520 info**

To adjust only the lease time globally for all scopes on all interfaces to a value of 4320 minutes (3 days), type:

**set global leasetime=4320**

Top of page

### set interface

Configures DHCP allocator parameters for the specified interface.

**Syntax**

**set interface** [**InterfaceName=**]*InterfaceName* [**mode=**]{**enable** | **disable**}

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ mode=]{ enable| disable}**

Required. Specifies whether DHCP allocation is enabled or disabled for the interface.

Top of page

### show global

Displays the DHCP allocator global configuration.

**Syntax**

**show global**

**Parameters**

none

Top of page

### show interface

Displays the DHCP allocator configuration for the specified interface.

**Syntax**

**show interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### uninstall

Removes the routing protocol under IP.

**Syntax**

**uninstall**

**Parameters**

none

Top of page

## Netsh Routing IP Dnsproxy commands

The following commands are available for use in the Netsh Routing IP Dnsproxy context.

To view the command syntax, click a command:



### set global

Sets global DNS proxy parameters.

**Syntax**

**set global** [**querytimeout=**]*Integer* [[**dnsmode=**]{**enable** | **disable**}] [[**loglevel=**]{**none** | **error** | **warn** | **info**}]

**Parameters**

**[ querytimeout=]** ***Integer***

Required. Specifies a time-out, in seconds, for proxied DNS queries.

**[ dnsmode=]{ enable| disable}**

Specifies whether DNS proxy operation is enabled or disabled.

**[ loglevel=]{ none| error| warn| info}**

Specifies which events should be logged. The **none** parameter specifies that no events related to DNS proxy should be logged. The **error** parameter specifies that only errors related to DNS proxy should be logged. The **warn** parameter specifies that only warnings related to DNS proxy should be logged. The **info** parameter specifies that all events related to DNS proxy should be logged.

**Examples**

To specify that queries should time out after 10 seconds, to enable DNS proxy operation, and to prevent logging of DNS proxy events, type:

**set global 10 enable none**

Top of page

## install

Installs the routing protocol under IP.

**Syntax**

**install**

**Parameters**

none

Top of page

## set interface

Configures DNS proxy parameters for an interface.

**Syntax**

**set interface** [**InterfaceName=**]*InterfaceName* [[**mode=**]{**enable** | **disable** | **default**}]

**Parameters**

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface for which to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ mode=]{ enable| disable| default}**

Specifies whether DNS proxy is enabled, disabled, or set to use the default for the specified interface. The **enable** parameter specifies that DNS proxy is enabled. The **disable** parameter specifies that DNS proxy is disabled. The **default** parameter specifies that DNS proxy is set according to the default for the interface.

Top of page

## show global

Displays the DNS proxy global configuration.

**Syntax**

**show global**

**Parameters**

none

Top of page

### show interface

Displays the DNS proxy configuration for the specified interface.

**Syntax**

**show interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=]** ***InterfaceName***

Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

### uninstall

Removes the routing protocol under IP.

**Syntax**

**uninstall**

**Parameters**

none

Top of page

## Netsh Routing IP NAT commands

The following commands are available for use in the Netsh Routing IP NAT context.

To view the command syntax, click a command:



### add addressmapping

Adds an IP address mapping to the network address translation (NAT) address pool for the specified interface.

**Syntax**

**add addressmapping** [**InterfaceName=**]*InterfaceName* [**public=**]*IPAddress* [**private=**]*IPAddress* [**inboundsessions=**] {**enable** | **disable**}

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to use network address translation. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ public=] *IPAddress***

Required. Specifies the IP address used on a public network.

**[ private=] *IPAddress***

Required. Specifies the IP address in the pooled range of addresses available for use on the private network.

**[ inboundsessions=]{ enable| disable}**

Required. Specifies whether inbound sessions are enabled or disabled. The **enable** parameter specifies that inbound sessions are enabled. The **disable** parameter specifies that inbound sessions are disabled.

**Examples**

To map the public IP address 10.11.11.1 to the private IP address 10.10.10.1 and to disable inbound sessions for this NAT table mapping, type:

**add addressmapping Local Area Connection 10.11.11.1 10.10.10.1 disable**

Top of page

## add addressrange

Adds an address range to the NAT address pool for the specified interface.

**Syntax**

**add addressrange** [**InterfaceName=**]*InterfaceName* [**start=**]*IPAddress* [**end=**]*IPAddress* [**mask=**]*SubnetMask*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface to whose address pool you want to add an address range.

**[ start=] *IPAddress***

Required. Specifies the starting IP address for the address range.

**[ end=] *IPAddress***

Required. Specifies the ending IP address for the address range.

**[ mask=] *SubnetMask***

Required. Specifies the IP subnet mask associated with the network range bounded by the start and end IP addresses.

Top of page

## add interface

Configures NAT on the specified interface.

**Syntax**

**add interface** [**InterfaceName=**]*InterfaceName* [[**mode=**]{**full** | **addressonly** | **private**}]

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface on which you want to configure NAT. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ mode=]{ full| addressonly| private}**

Specifies whether the interface uses full, addressonly, or private mode.

Top of page

### add portmapping

Adds a protocol port mapping on the NAT interface.

#### Syntax

**add portmapping** [**InterfaceName=**]*InterfaceName* [**proto=**]{**tcp** | **udp**} [**publicip=**]{*IPAddress* | **0.0.0.0**} [**publicport=**]*Integer* [**privateip=**]*IPAddress* [**privateport=**]*Integer*

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to add a port mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ proto=]{ tcp| udp}**

Required. Specifies whether the protocol type is set to TCP or UDP.

**[ publicip=]{ *IPAddress*| 0.0.0.0}**

Required. Specifies an external IP address on the public network or 0.0.0.0 to indicate any IP address not specified within the private network address range.

**[ publicport=] *Integer***

Required. Specifies the public protocol port by using a number from 0 to 9999.

**[ privateip=] *IPAddress***

Required. Specifies an IP address within the private network range.

**[ privateport=] *Integer***

Required. Specifies the private protocol port by using a number from 0 to 9999.

Top of page

### delete addressmapping

Deletes an address mapping from the NAT address pool for the specified interface.

#### Syntax

**delete addressmapping** [**InterfaceName=**]*InterfaceName* [[**public=**]*IPAddress*]

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to delete an address mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ public=] *IPAddress***

Specifies the IP address to be deleted. If you do not specify an address, all address mappings from the NAT address pool are deleted from the interface.

Top of page

### delete addressrange

Deletes an address range from the NAT address pool for the specified interface.

### Syntax

**delete addressrange** [**InterfaceName=**]*InterfaceName* [**start=**]*IPAddress*

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface from which you want to delete an address range. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ start=] *IPAddress***

Required. Specifies the starting IP address of the range you want to delete.

Top of page

## delete interface

Removes network address translation (NAT) from the specified interface.

### Syntax

**delete interface** [**InterfaceName=**]*InterfaceName*

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to remove the use of NAT. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

## delete portmapping

Deletes a protocol port mapping from the specified NAT-enabled interface.

### Syntax

**delete portmapping** [**InterfaceName=**]*InterfaceName* [[**proto=**]{**tcp** | **udp**}] [**publicip=**]{*IPAddress* | **0.0.0.0**}] [[**publicport=**]*Integer*] [[**privateip=**]*IPAddress*] [**privateport=**]*Integer*]

### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to delete a port mapping. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ proto=]{ tcp| udp}**

Specifies the protocol type. The **tcp** parameter specifies that the protocol type is Transmission Control Protocol. The **udp** parameter specifies that the protocol type is User Datagram Protocol.

**[ publicip=]{ *IPAddress*| 0.0.0.0}**

Specifies an external IP address on the public network or 0.0.0.0 to indicate any IP address not specified within the private network address range.

**[ publicport=] *Integer***

Specifies the public protocol port by using a number from 0 to 9999.

**[ privateip=] *IPAddress***

Specifies an IP address within the private network range.

**[ privateport=] *Integer***

Specifies the private protocol port by using a number from 0 to 9999.

Top of page

### set global

Sets global parameters for NAT.

#### Syntax

**set global** {[[**tcptimeoutmins=**]*Integer*] | [[**udptimeoutmins=**]*Integer*] | [[**LogLevel**=]{**none** | **error** | **warn** | **info**}]}

#### Parameters

**[ tcptimeoutmins=] *Integer***

Specifies, in minutes, the time-out value for TCP mappings.

**[ udptimeoutmins=] *Integer***

Specifies, in minutes, the time-out value for UDP mappings.

**[ LogLevel=]{ none| error| warn| info}**

Specifies which events should be logged. The **none** parameter specifies that no events related to NAT should be logged. The **error** parameter specifies that only errors related to NAT should be logged. The **warn** parameter specifies that only warnings related to NAT should be logged. The **info** parameter specifies that all events related to NAT should be logged.

Top of page

### set interface

Configure NAT parameters for the specified interface.

#### Syntax

**set interface** [**InterfaceName=**]*InterfaceName* [**mode=**]{*Full* | *AddressOnly* | *Private*}

#### Parameters

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to configure parameters. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

**[ mode=]{ *Full*| *AddressOnly*| *Private*}**

Required. Specifies whether the interface should be enabled for full, addressonly, or private mode.

Top of page

### show global

Displays network address translation (NAT) global configuration.

#### Syntax

**show global**

#### Parameters

none

Top of page

### show interface

Displays network address translation (NAT) configuration for the specified interface.

#### Syntax

**show interface** [**InterfaceName=**]*InterfaceName*

**Parameters**

**[ InterfaceName=] *InterfaceName***

Required. Specifies, by name, the interface for which you want to display information. The *InterfaceName* parameter must match the name of the interface as specified in Network Connections. If *InterfaceName* contains spaces, use quotation marks around the text (for example, **"***Interface Name***"**).

Top of page

## Netsh Routing example

To enable router discovery for the interface named Local Area Connection, specifying a minimum interval of 7 minutes, a maximum interval of 10 minutes, a life of 30 minutes for router discovery packets, and a zero (0) or no preference level, type the following from the netsh routing prompt:

**add interface "Local Area Connection" enable 7 10 30 0**

## Formatting legend

| Format | Meaning |
|---|---|
| *Italic* | Information that the user must supply |
| **Bold** | Elements that the user must type exactly as shown |
| Ellipsis (...) | Parameter that can be repeated several times in a command line |
| Between brackets ([]) | Optional items |
| Between braces ({}); choices separated by pipe (\|). Example: {even\|odd} | Set of choices from which the user must choose only one |
| **Courier font** | Code or program output |

**Related Links**


- Netsh commands for AAAA
- Netsh commands for DHCP
- Netsh diagnostic (diag) commands
- Netsh commands for Interface IP
- Netsh commands for remote access (ras)
- Netsh commands for WINS
- Command-line reference A-Z
- Command shell overview