

## Netsh commands for Interface IPv6

Updated: January 21, 2005

### Netsh commands for Interface IPv6

The Netsh commands for Interface IPv6 provide a command-line tool that you can use to query and configure IPv6 interfaces, address, caches, and routes.

In addition, the Interface IPv6 context of **netsh** has a subcontext for 6to4. You can use the commands in the **netsh interface IPv6 6to4** context to configure or display the configuration of the 6to4 service on either a 6to4 host or a 6to4 router.

You can run these commands from the Windows Server™ 2003 family command prompt or from the command prompt for the **netsh interface IPv6** context. For these commands to work at the Windows Server™ 2003 family command prompt, you must type **netsh interface ipv6** before typing commands and parameters as they appear in the syntax below. There might be functional differences between **netsh** context commands on Windows 2000 and the Windows Server™ 2003 family.

To view help for a command at the command prompt, type *CommandName/?*, where *CommandName* is the name of the command.

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

To view the command syntax, click a command:

- [6to4](#)
- [add 6over4tunnel](#)
- [add address](#)
- [add dns](#)
- [add prefixpolicy](#)
- [add route](#)
- [add v6v4tunnel](#)
- [delete address](#)
- [delete destinationcache](#)
- [delete dns](#)
- [delete interface](#)
- [delete neighbors](#)
- [delete prefixpolicy](#)
- [delete route](#)
- [dump](#)
- [install](#)
- [isatap](#)
- [renew](#)
- [reset](#)
- [set address](#)

- [set global](#)
- [set interface](#)
- [set mobility](#)
- [set prefixpolicy](#)
- [set privacy](#)
- [set route](#)
- [set state](#)
- [show address](#)
- [show bindingcacheentries](#)
- [show destinationcache](#)
- [show dns](#)
- [show global](#)
- [show interface](#)
- [show joins](#)
- [show mobility](#)
- [show neighbors](#)
- [show prefixpolicy](#)
- [show privacy](#)
- [show routes](#)
- [show siteprefixes](#)
- [uninstall](#)

## 6to4

Specifies that the **6to4** context of **netsh interface IPv6 6to4** is used.

### Syntax

**6to4**

[↑ Top of page](#)

## add 6over4tunnel

Creates a 6over4 interface using the specified IPv4 address.

### Syntax

```
add 6over4tunnel [[interface=]String] [localaddress=]IPv4Address [[store=]{active | persistent}]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[ localaddress=] IPv4Address**

Required. Specifies the IPv4 address that is encapsulated.

**[[ store=]{ active| persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command creates a 6over4 interface using the IPv4 address 10.1.1.1 on the interface named "Private."

```
add 6over4tunnel "Private" 10.1.1.1
```

[↑ Top of page](#)

## add address

Adds an IPv6 address to a specified interface. Time values can be expressed in days (d), hours (h), minutes (m), and seconds (s). For example, 2d represents two days.

### Syntax

```
add address [[interface=]String] [address=]IPv6Address [[type=]{unicast | anycast}] [[validlifetime=]{Integer | infinite}] [[preferredlifetime=]{Integer | infinite}] [[store=]{active | persistent}]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[ address=] IPv6Address**

Required. Specifies the IPv6 address to add.

**[[ type=]{ unicast| anycast}]**

Specifies whether a unicast address (**unicast**) or an anycast address (**anycast**) is added. The default selection is **unicast**.

**[[ validlifetime=]{ Integer| infinite}]**

Specifies the lifetime over which the address is valid. The default value is **infinite**.

**[[ preferredlifetime=]{ Integer| infinite}]**

Specifies the lifetime over which the address is preferred. The default value is **infinite**.

**[[ store=]{ active| persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command adds the IPv6 address FE80::2 to the interface named "Private."

```
add address "Private" FE80::2
```

[↑ Top of page](#)

## add dns

Adds a new DNS server IP address to the statically-configured list of DNS servers for the specified interface.

### Syntax

```
add dns [interface=]String [address=]IPAddress [[index=]Integer]
```

### Parameters

**[ interface=] String**

Required. Specifies, by name, which interface will have a DNS server IP address added to its list of DNS server IP addresses.

**[ address=] IPAddress**

Required. Specifies the IPv6 address of the DNS server to add to the list.

**[[ index=] Integer]**

Specifies the position on the statically-configured list in which to place the DNS server IP address specified in **address**. By default, the DNS server IP address is added to the end of the list.

### Remarks

If an index is specified, the DNS server is placed in that position in the list.

### Examples

In the first example command, a DNS server with the IPv6 address FEC0:0:0:FFFF::1 is added to the list of DNS server IP addresses for the interface named "Local Area Connection." In the second example, a DNS server with the IPv6 address FEC0:0:0:FFFF::2 is added at index 2 as the second server on the list of servers for the interface named "Local Area Connection".

```
add dns "Local Area Connection" FEC0:0:0:FFFF::1
```

```
add dns "Local Area Connection" FEC0:0:0:FFFF::2 index=2
```

[↑ Top of page](#)

### add prefixpolicy

Adds a source and destination address selection policy for a specified prefix.

#### Syntax

```
add prefixpolicy [prefix=]IPv6Address/Integer [precedence=]Integer [label=]Integer [[store=]{active | persistent}]
```

#### Parameters

**[ prefix= ] IPv6Address/Integer**

Required. Specifies the prefix for which to add a policy in the policy table. *Integer* specifies the prefix length.

**[ precedence= ] Integer**

Required. Specifies the precedence value used for sorting destination addresses in the policy table.

**[ label= ] Integer**

Required. Specifies the label value that allows for policies that require a specific source address prefix for use with a destination address prefix.

**[[ store= ]{ active | persistent }]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command adds a prefix policy for prefix ::/96, with a precedence value of 3 and a label value of 4.

```
add prefixpolicy ::/96 3 4
```

[↑ Top of page](#)

### add route

Adds a route for a specified prefix. Time values can be expressed in days (d), hours (h), minutes (m), and seconds (s). For example, 2d represents two days.

#### Syntax

```
add route [prefix=]IPv6Address/Integer [[interface=]String] [[nexthop=]IPv6Address] [[siteprefixlength=]Integer] [[metric=]Integer] [[publish=]{no | yes | immortal}] [[validlifetime=]{Integer | infinite}] [[preferredlifetime=]{Integer | infinite}] [[store=]{active | persistent}]
```

#### Parameters

**[ prefix= ] IPv6Address/Integer**

Required. Specifies the prefix for which to add a route. *Integer* specifies the prefix length.

**[[ interface= ] String]**

Specifies an interface name or index.

**[[ nexthop= ] IPv6Address]**

Specifies the gateway address, if the prefix is not on-link.

**[[ siteprefixlength=] *Integer*]**

Specifies the prefix length for the entire site, if the prefix is not on-link.

**[[ metric=] *Integer*]**

Specifies the route metric.

**[[ publish=]{ no | yes | immortal}]**

Specifies whether routes are advertised (**yes**), advertised with an infinite lifetime (**immortal**), or not advertised (**no**) in Route Advertisements. The default selection is **no**.

**[[ validlifetime=]{ *Integer* | infinite}]**

Specifies the lifetime over which the route is valid. The default value is **infinite**.

**[[ preferredlifetime=]{ *Integer* | infinite}]**

Specifies the lifetime over which the route is preferred. The default value is **infinite**.

**[[ store=]{ active | persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command adds a route on the interface named "Internet" with a prefix of 3FFE:: and a prefix length of 16 bits (3FFE::/16). The **nexthop** value is FE80::1.

```
add route 3FFE::/16 "Internet" FE80::1
```

[↑ Top of page](#)

## add v6v4tunnel

Creates an IPv6-in-IPv4 tunnel.

### Syntax

```
add v6v4tunnel [[interface=] String] [localaddress=] IPv4Address [remoteaddress=] IPv4Address
[[neighbordiscovery=]{ enabled | disabled}] [[store=]{ active | persistent}]
```

### Parameters

**[[ interface=] *String*]**

Specifies an interface name or index.

**[ localaddress=] *IPv4Address***

Required. Specifies the IPv4 address of the local tunnel endpoint.

**[ remoteaddress=] *IPv4Address***

Required. Specifies the IPv4 address of the remote tunnel endpoint.

**[[ neighbordiscovery=]{ enabled | disabled}]**

Specifies whether Neighbor Discovery is enabled (**enabled**) or disabled (**disabled**) on the interface. The default selection is **disabled**.

**[[ store=]{ active | persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command creates an IPv6-in-IPv4 tunnel between the local address 10.0.0.1 and the remote address 192.168.1.1 on the interface "Private."

```
add v6v4tunnel "Private" 10.0.0.1 192.168.1.1
```

[↑ Top of page](#)

## delete address

## Syntax

```
delete address [[interface=]String] [address=]IPv6Address [[store=]{active | persistent}]
```

Modifies an IPv6 address on a specified interface.

## Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[ address=] IPv6Address**

Required. Specifies the IPv6 address to delete.

**[[ store=]{ active| persistent}]**

Specifies whether the deletion lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

## Examples

This example command deletes the address FE80::2 from the interface named "Private".

```
delete address "Private" FE80::2
```

[↑ Top of page](#)

## delete destinationcache

Clears the destination cache. If an interface is specified, clears the cache only on that interface. If an address is also specified, deletes only that destination cache entry.

## Syntax

```
delete destinationcache [[interface=]String] [[address=]IPv6Address]
```

## Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ address=] IPv6Address]**

Specifies the address of the destination.

## Remarks

When no parameters are specified, all entries in the destination caches for all interfaces are deleted.

## Examples

This example command deletes the destination cache for the interface named "Private."

```
delete destinationcache "Private"
```

[↑ Top of page](#)

## delete dns

Deletes statically configured DNS server IPv6 addresses for a specific interface.

## Syntax

```
delete dns [interface=]String [[address=]{ IPv6Address | all}]
```

## Parameters

**[ interface=] String**

Required. Specifies the interface, by name, for which you want to remove a DNS server from the list of DNS servers.

**[[ address=]{ IPv6Address | all}]**

Specifies the DNS server IPv6 address to delete. If **all** is specified, all DNS server IPv6 addresses on the list for the interface are deleted.

### Examples

In the first example command, the DNS server IPv6 address FEC0:0:0:FFFF::1 is deleted from the list of addresses for the connection named "Local Area Connection." In the second example command, all DNS server IPv6 addresses are deleted for the connection named "Local Area Connection."

```
delete dns "Local Area Connection" FEC0:0:0:FFFF::1
```

```
delete dns "Local Area Connection" all
```

[↑ Top of page](#)

### delete interface

Deletes a specified interface from the IPv6 stack.

### Syntax

```
delete interface [[interface=]String] [[store=]{ active | persistent}]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ store=]{ active | persistent}]**

Specifies whether the deletion lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command deletes the interface named "Private."

```
delete interface "Private"
```

[↑ Top of page](#)

### delete neighbors

Specifies that all entries in the neighbor cache are deleted. If an interface is specified, clears the cache only on that interface. If an address is also specified, deletes only that neighbor cache entry.

### Syntax

```
delete neighbors [[interface=]String] [[address=]IPv6Address]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ address=] IPv6Address]**

Specifies the address of the neighbor.

### Examples

This example command removes all entries from the neighbor cache on the interface named "Private."

```
delete neighbors "Private"
```

[↑ Top of page](#)

### delete prefixpolicy

Deletes the source and destination address selection policy for a specified prefix.

### Syntax

```
delete prefixpolicy [prefix=]IPv6Address/ Integer [[store=]{ active | persistent}]
```

### Parameters

[ **prefix=**] IPv6Address/ Integer

Required. Specifies the prefix (*IPv6Address*) and prefix length (*Integer*) to delete from the policy table.

[[ **store=**]{ **active**| **persistent**}]

Specifies whether the deletion lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command deletes the prefix `::/96` from the policy table.

```
delete prefixpolicy ::/96
```

[↑ Top of page](#)

## delete route

Deletes an IPv6 route.

### Syntax

```
delete route [prefix=]IPv6Address/ Integer [[interface=]String] [[nexthop=]IPv6Address] [[store=]{ active | persistent}]
```

### Parameters

[ **prefix=**] IPv6Address/ Integer

Required. Specifies the prefix of the route to delete.

[[ **interface=**] *String*]

Specifies an interface name or index.

[[ **nexthop=**] *IPv6Address*]

Specifies the gateway address, if the prefix is not on-link.

[[ **store=**]{ **active**| **persistent**}]

Specifies whether the deletion lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command deletes the route with the prefix `3FFE::/16` and the gateway `FE80::1` from the interface named "Internet."

```
delete route 3FFE::/16 "Internet" FE80::1
```

[↑ Top of page](#)

## dump

Dumps the network adapter IPv6 configuration to the command prompt window when run within the netsh context. When used in a batch file or script, output can be saved in a text file.

### Syntax

```
netsh interface ipv6 dump > [PathAndFileName]
```

### Parameters

[ *PathAndFileName*]



Specifies both the location where the file is saved and the name of the destination file to which the configuration is saved.

### Remarks

- After file output is obtained, you can use the **netsh exec** command to either configure another computer with the same IPv6 configuration or to restore the original configuration on the same computer.
- All IPv6 configuration information is saved with the dump command. For example, if an ISATAP or a 6to4 configuration is defined on an interface, the **dump** command saves these settings in the text file.

### Examples

In the first example, the command is run manually at the **netsh interface ipv6** context of a command prompt. The IPv6 configuration is displayed in the command prompt window, and can be copied and pasted into a text file. In the second example, the **dump** command is run in a batch file, and the configuration is saved to a text file named Ipv6\_conf.txt at the location C:\Temp.

### dump

```
netsh interface ipv6 dump > C:\temp\ipv6_conf.txt
```

[↑ Top of page](#)

### install

Installs IPv6.

### Syntax

```
install
```

[↑ Top of page](#)

### isatap

Specifies that the **isatap** context of **netsh interface IPv6 isatap** is used.

### Syntax

```
isatap
```

### Remarks

- Intrasite Automatic Tunnel Addressing Protocol (ISATAP) is used for communication between IPv6 and IPv4 nodes within an IPv4 site. It is described in the Internet draft titled "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)" (draft-ietf-ngtrans-isatap-00.txt). For more information, see [IPv6 traffic between nodes on different subnets of an IPv4 internetwork](http://technet2.microsoft.com/WindowsServer/en/library/4a5a3ee3-4e42-4002-9717-28b82ef64c921033.msp) [http://technet2.microsoft.com/WindowsServer/en/library/4a5a3ee3-4e42-4002-9717-28b82ef64c921033.msp] .

[↑ Top of page](#)

### renew

Restarts IPv6 interfaces.

### Syntax

```
renew [[interface=]String]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

### Examples

renew "Private"

[↑ Top of page](#)

## reset

Resets the IPv6 configuration state.

### Syntax

reset

[↑ Top of page](#)

## set address

Modifies an IPv6 address on a specified interface. Time values can be expressed in days (d), hours (h), minutes (m), and seconds (s). For example, 2d represents two days.

### Syntax

```
set address [[interface=]String] [address=]IPv6Address [[type={unicast | anycast}] [[validlifetime={ Integer | infinite}] [[preferredlifetime={ Integer | infinite}] [[store={ active | persistent}]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[ address=] IPv6Address**

Required. Specifies the IPv6 address to modify.

**[[ type={ unicast | anycast}]**

Specifies whether the address is marked as a unicast address (**unicast**) or as an anycast address (**anycast**). The default selection is **unicast**.

**[[ validlifetime={ Integer | infinite}]**

Specifies the lifetime over which the address is valid. The default value is **infinite**.

**[[ preferredlifetime={ Integer | infinite}]**

Specifies the lifetime over which the address is preferred. The default value is **infinite**.

**[[ store={ active | persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command sets the address FE80::2 on the interface named "Private" as an anycast address.

```
set address "Private" FE80::2 anycast
```

[↑ Top of page](#)

## set global

Modifies global configuration parameters.

### Syntax

```
set global [[defaultcurhoplimit=]Integer] [neighborcachelimit=]Integer [[routecachelimit=]Integer] [[reassemblylimit=]Integer] [[store={ active | persistent}]
```

### Parameters

**[[ defaultcurhoplimit=] Integer]**

Specifies the default hop limit of packets sent.

**[[ neighborcachelimit=] *Integer***

Required. Specifies the maximum number of neighbor cache entries.

**[[ routecachelimit=] *Integer***

Specifies the maximum number of route cache entries.

**[[ reassemblylimit=] *Integer***

Specifies the maximum size of the reassembly buffer.

**[[ store=]{ *active* | *persistent* }**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

## Examples

This example command sets global parameters for all IPv6-enabled interfaces on the computer. The default hop limit is set to 32, the maximum number of neighbor cache entries is set to 100, and the maximum number of route cache entries is 100,000.

```
set global 32 100 100000
```

[↑ Top of page](#)

## set interface

Modifies interface configuration parameters.

### Syntax

```
set interface [[interface=] String] [[forwarding=]{ enabled | disabled }] [[advertise=]{ enabled | disabled }] [[mtu=] Integer] [[siteid=] Integer] [[metric=] Integer] [[firewall=]{ enabled | disabled }] [[siteprefixlength=] Integer] [[store=]{ active | persistent }
```

### Parameters

**[[ interface=] *String***

Specifies an interface name or index.

**[[ forwarding=]{ *enabled* | *disabled* }**

Specifies whether packets arriving on this interface can be forwarded to other interfaces. The default selection is **disabled**.

**[[ advertise=]{ *enabled* | *disabled* }**

Specifies whether Router Advertisements are sent on this interface. The default selection is **disabled**.

**[[ mtu=] *Integer***

Specifies the MTU of this interface. The default MTU is the natural MTU of the link.

**[[ siteid=] *Integer***

Specifies the site scope zone identifier.

**[[ metric=] *Integer***

Specifies the interface metric, which is added to route metrics for all routes over the interface.

**[[ firewall=]{ *enabled* | *disabled* }**

Specifies whether to operate in firewall mode.

**[[ siteprefixlength=] *Integer***

Specifies the default length of the global prefix for the entire site.

**[[ store=]{ *active* | *persistent* }**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

## Examples

This example command sets the interface with the name "Private," with a siteid of 2 and a metric of 2. All other parameter values are left at the default values.

**set interface "Private" siteid=2 metric=2**

[↑ Top of page](#)

## set mobility

Modifies mobility configuration parameters.

### Syntax

```
set mobility [[security=]{enabled | disabled}] [[bindingcachelimit=]Integer] [[correspondentnode=]enabled | disabled] [[store=]{active | persistent}]
```

### Parameters

**[[ security=]{ enabled| disabled}]**

Specifies whether binding updates must be secured.

**[[ bindingcachelimit=] Integer]**

Specifies the maximum number of binding cache entries.

**[[ correspondentnode=] enabled| disabled]**

Specifies whether Correspondent Node functionality is set to **enabled** or the default of **disabled**.

**[[ store=]{ active| persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

```
set mobility security=disabled bindingcachelimit=1000 corr=enabled
```

[↑ Top of page](#)

## set prefixpolicy

Modifies a source and destination address selection policy for a specified prefix.

### Syntax

```
set prefixpolicy [prefix=]IPv6Address/Integer [precedence=]Integer [label=]Integer [[store=]{active | persistent}]
```

### Parameters

**[ prefix=] IPv6Address/Integer**

Required. Specifies the prefix for which to add a policy in the policy table. *Integer* specifies the prefix length.

**[ precedence=] Integer**

Required. Specifies the precedence value used for sorting destination addresses in the policy table.

**[ label=] Integer**

Required. Specifies the label value that allows for policies that require a specific source address prefix for use with a destination address prefix.

**[[ store=]{ active| persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command sets a policy in the policy table for the prefix `::/96`, with a precedence value of 3 and a label value of 4.

```
set prefixpolicy ::/96 3 4
```

[↑ Top of page](#)

## set privacy

Modifies parameters related to temporary address generation. If **randomtime=** is specified, **maxrandomtime=** is not used. Time values can be expressed in days (d), hours (h), minutes (m), and seconds (s). For example, 2d represents two days.

### Syntax

```
set privacy [[state={enabled | disabled}] [[maxdadattempts=] Integer] [[maxvalidlifetime=] Integer]
[[maxpreferredlifetime=] Integer] [[regeneratetime=] Integer] [[maxrandomtime=] Integer] [[randomtime=] Integer]
[[store={active | persistent}]
```

### Parameters

**[[ state=]{ enabled| disabled}]**

Specifies whether temporary addresses are enabled.

**[[ maxdadattempts=] Integer]**

Specifies the number of duplicate address detection attempts made. The default value is 5.

**[[ maxvalidlifetime=] Integer]**

Specifies the maximum lifetime over which a temporary address is valid. The default value is 7d (seven days).

**[[ maxpreferredlifetime=] Integer]**

Specifies the maximum lifetime over which an anonymous is preferred. The default value is 1d (one day).

**[[ regeneratetime=] Integer]**

Specifies the duration of time that elapses when a new address is generated prior to deprecating a temporary address. The default value is 5s (five seconds).

**[[ maxrandomtime=] Integer]**

Specifies the upper limit to use when computing a random delay at boot. The default value is 10m (ten minutes).

**[[ randomtime=] Integer]**

Specifies a time value to use, instead of a value generated at boot.

**[[ store=]{ active| persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

[↑ Top of page](#)

## set route

Modifies route parameters. Time values can be expressed in days (d), hours (h), minutes (m), and seconds (s). For example, 2d represents two days.

### Syntax

```
set route [prefix=] IPv6Address/ Integer [[interface=] String] [[nexthop=] IPv6Address] [[siteprefixlength=] Integer]
[[metric=] Integer] [publish={no | yes | immortal}] [[validlifetime=]{ Integer | infinite}] [[preferredlifetime=]
{ Integer | infinite}] [[store={active | persistent}]
```

### Parameters

**[ prefix=] IPv6Address/ Integer**

Required. Specifies the prefix (*IPv6Address*) and prefix length (*Integer*) of the route to modify.

**[[ interface=] String]**

Specifies an interface name or index.

**[[ nexthop=] IPv6Address]**

Specifies the gateway address, if the prefix is not on-link.

**[[ siteprefixlength=] Integer]**

Specifies the prefix length for the entire site, if the prefix is not on-link.

**[[ metric=] Integer]**

Specifies the route metric.

**[[ publish=]{ no | yes | immortal}]**

Specifies whether routes are advertised (**yes**), advertised with an infinite lifetime (**immortal**), or not advertised (**no**) in Route Advertisements. The default selection is **no**.

**[[ validlifetime=]{ Integer | infinite}]**

Specifies the lifetime over which the route is valid. The default value is **infinite**.

**[[ preferredlifetime=]{ Integer | infinite}]**

Specifies the lifetime over which the route is preferred. The default value is **infinite**.

**[[ store=]{ active | persistent}]**

Specifies whether the change lasts only until the next boot (**active**) or is persistent (**persistent**). The default selection is **persistent**.

### Examples

This example command sets a route on the interface named "Internet." The route prefix is 3FFE::, and has a length of 16 bits. The gateway address, defined by the **nexthop=** parameter, is FE80::1.

```
set route 3FFE::/16 "Internet" FE80::1
```

[↑ Top of page](#)

### set state

Enables or disables IPv4 compatibility. The default value for all parameters is disabled.

### Syntax

```
set state [[6over4=]{ enabled | disabled | default}] [[v4compat=]{ enabled | disabled | default}]
```

### Parameters

**[[6over4=]{ enabled | disabled | default}]**

Specifies whether 6over4 interfaces are created. To both disable and delete 6over4 compatible interfaces, specify **default**. To disable 6over4 compatible interfaces without deleting them, specify **disabled**.

**[[ v4compat=]{ enabled | disabled | default}]**

Specifies whether IPv4 compatible interfaces are created. To both disable and delete IPv4 compatible interfaces, specify **default**. To disable IPv4 compatible interfaces without deleting them, specify **disabled**.

### Examples

In the first example command, IPv4-compatible addresses are disabled, and any previously existing interfaces are deleted. In the second example command, IPv4-compatible addresses are enabled

```
set state default
```

```
set state 6over4=disabled v4compat=enabled
```

[↑ Top of page](#)

### show address

Displays all IPv6 addresses, or all addresses on a specified interface.

### Syntax

```
show address [[interface=]String] [[level=]{ normal | verbose}] [[store=]{ active | persistent}]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ level=]{ normal | verbose}]**

Specifies whether one line per interface is displayed (**normal**) or additional information is displayed for each interface (**verbose**). When no interface is specified, the default selection is **normal**. When an interface is specified, the default

selection is **verbose**.

```
[[ store=]{ active| persistent}]
```

Specifies whether active (**active**) or persistent (**persistent**) addresses are displayed. The default selection is **active**.

[↑ Top of page](#)

## show bindingcacheentries

Displays all binding cache entries.

### Syntax

```
show bindingcacheentries
```

[↑ Top of page](#)

## show destinationcache

Displays destination cache entries. If an interface is specified, displays the cache only on that interface. If an address is also specified, displays only that destination cache entry.

### Syntax

```
show destinationcache [[interface=]String] [[address=]IPv6Address]
```

### Parameters

```
[[ interface=] String]
```

Specifies an interface name or index.

```
[[ address=] IPv6Address]
```

Specifies the address of the destination.

[↑ Top of page](#)

## show dns

Displays the DNS server configuration for a specific interface or interfaces.

### Syntax

```
show dns [[interface=]String]
```

### Parameters

```
[[ interface=] String]
```

Specifies the interface, by name, for which you want to display configured DNS server IPv6 addresses. If no interface is specified, servers for all interfaces are displayed.

### Examples

In this example command, DNS server IPv6 addresses configured on the "Local Area Connection" interface are displayed.

```
show dns "Local Area Connection"
```

[↑ Top of page](#)

## show global

Displays global configuration parameters.

### Syntax

```
show global [[store=]{active | persistent}]
```

## Parameters

**[[ store=]{ active | persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) information is displayed. The default selection is **active**.

[↑ Top of page](#)

## show interface

Displays information about all interfaces, or about a specified interface.

### Syntax

**show interface** **[[interface=]String]** **[[level=]{ normal | verbose}]** **[[store=]{ active | persistent}]**

## Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ level=]{ normal | verbose}]**

Specifies whether one line per interface is displayed (**normal**) or additional information is displayed for each interface (**verbose**). When no interface is specified, the default selection is **normal**. When an interface is specified, the default selection is **verbose**.

**[[ store=]{ active | persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) interfaces are displayed. The default selection is **active**.

[↑ Top of page](#)

## show joins

Displays all IPv6 multicast addresses, or all multicast addresses on a specified interface.

### Syntax

**show joins** **[[interface=]String]** **[[level=]{ normal | verbose}]**

## Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ level=]{ normal | verbose}]**

Specifies whether one line per interface is displayed (**normal**) or additional information is displayed for each interface (**verbose**). When no interface is specified, the default selection is **normal**. When an interface is specified, the default selection is **verbose**.

[↑ Top of page](#)

## show mobility

Displays mobility configuration parameters.

### Syntax

**show mobility** **[[store=]{ active | persistent}]**

## Parameters

**[[ store=]{ active | persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) information is displayed. The default selection is **active**.

[↑ Top of page](#)



## show neighbors

Displays neighbor cache entries. If an interface is specified, displays only the cache on that interface. If an address is also specified, displays only that neighbor cache entry.

### Syntax

```
show neighbors [[interface=]String] [[address=]IPv6Address]
```

### Parameters

**[[ interface=] String]**

Specifies an interface name or index.

**[[ address=] IPv6Address]**

Specifies the address of the neighbor.

[↑ Top of page](#)

## show prefixpolicy

Displays prefix policy table entries used in source and destination address selection.

### Syntax

```
show prefixpolicy [[store=]{active | persistent}]
```

### Parameters

**[[ store=]{ active| persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) information is displayed. The default selection is **active**.

[↑ Top of page](#)

## show privacy

Displays privacy configuration parameters.

### Syntax

```
show privacy [[store=]{active | persistent}]
```

### Parameters

**[[ store=]{ active| persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) information is displayed. The default selection is **active**.

[↑ Top of page](#)

## show routes

Displays route table entries.

### Syntax

```
show routes [[level=]{normal | verbose}] [[store=]{active | persistent}]
```

### Parameters

**[[ level=]{ normal| verbose}]**

Specifies whether only normal routes (**normal**) or routes used for loopback (**verbose**) are displayed. The default selection is **normal**.

**[[ store=]{ active| persistent}]**

Specifies whether active (**active**) or persistent (**persistent**) routes are displayed. The default selection is **active**.

[↑ Top of page](#)

## show siteprefixes

Displays the site prefix table.

### Syntax

**show siteprefixes**

[↑ Top of page](#)

## uninstall

Uninstalls IPv6.

### Syntax

**uninstall**

[↑ Top of page](#)

## Netsh interface IPv6 6to4

You can use the following commands in the **netsh interface IPv6 6to4** context to display the configuration of or configure the 6to4 service on either a 6to4 host or a 6to4 router. For more information about using the 6to4 service, see [IPv6 traffic between nodes in different sites across the Internet \(6to4\)](#) [<http://technet2.microsoft.com/WindowsServer/en/library/6ecf3d92-a57c-41b1-be9e-03a43331f2b71033.mspx>] .

For more information about Netsh, see [Netsh overview](#) [<http://technet2.microsoft.com/WindowsServer/en/library/61427fbd-de1f-4c8a-b613-321f7a3cca6a1033.mspx>] .

- [set interface](#)
- [show interface](#)
- [set relay](#)
- [show relay](#)
- [set routing](#)
- [show routing](#)
- [set state](#)
- [show state](#)
- [reset](#)

## set interface

Configures the 6to4 service on an interface.

### Syntax

**set interface** [**name=**] *InterfaceName* [[**routing=**] {**enabled** | **disabled** | **default**}]

### Parameters

[ **name=**] *InterfaceName*

Required. Specifies the name of the interface for which you want to set 6to4 service configuration. *InterfaceName* must match the name of the interface specified in Network Connections. If *InterfaceName* contains any spaces, it must be enclosed in quotes.

[[ **routing=**] { **enabled**| **disabled**| **default**}]

Specifies whether the forwarding of 6to4 packets received on the interface is enabled, disabled, or set to its default value.

#### Remarks

- This command enables, disables, or sets to default the routing behavior of the 6to4 service on a specified interface.

The default setting for the **routing=** parameter is **enabled**, which enables routing on private interfaces if Internet Connection Sharing (ICS) is used.

#### Note

- Internet Connection Sharing and Network Bridge are not included in Windows Server 2003, Web Edition; Windows Server 2003, Datacenter Edition; and the Itanium-based versions of the original release of the Windows Server 2003 operating systems.

[↑ Top of page](#)

## show interface

Displays the 6to4 service routing configuration on all interfaces, or on a specified interface.

#### Syntax

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

#### Parameters

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

Specifies the name of the interface for which you want to display the 6to4 service configuration. *InterfaceName* must match the name of the interface specified in Network Connections. If *InterfaceName* contains any spaces, it must be enclosed in quotes.

#### Remarks

- If an interface name is not specified, the 6to4 configuration for all interfaces is displayed.

[↑ Top of page](#)

## set relay

Configures the name of the 6to4 relay router for the 6to4 service. Additionally, specifies how often the name is resolved and the state of the relay component for the 6to4 service.

#### Syntax

**set relay** **[[name=]** { *RelayDNSName* | **default** } **[[state=]** { **enabled** | **disabled** | **automatic** | **default** } **[[interval=]** { *ResInterval* | **default** }

#### Parameters

**[[ name=]** { *RelayDNSName* | **default** }

Specifies either the fully qualified domain name (FQDN) of a 6to4 relay router on the IPv4 Internet (*RelayDNSName*), or sets the relay name to its default value of 6to4.ipv6.microsoft.com (**default**).

**[[ state=]** { **enabled** | **disabled** | **automatic** | **default** }

Specifies whether the state of the relay component for the 6to4 service is enabled, disabled, automatically enabled if a public IPv4 address is configured, or set to its default value.

**[[ interval=]** { *ResInterval* | **default** }

Specifies how often the name of the relay router is resolved in minutes (*ResInterval*) or sets the resolution interval to its default value of 1440 minutes (**default**).

#### Remarks

- The 6to4 relay router is a router that provides an access point between the IPv4 Internet and the 6bone (the native IPv6 portion of the Internet). In order to access 6bone resources from a 6to4 router, the 6to4 router encapsulates 6to4 traffic with an IPv4 header and sends it to the IPv4 address of the 6to4 relay router. The 6to4 relay router removes the IPv4 header and forwards the traffic to the 6bone. For return traffic, the 6to4 relay router encapsulates IPv6 traffic and sends it to the 6to4 router at the 6to4 host's site.

- The default name of the 6to4 relay router is 6to4.ipv6.microsoft.com.
- The default state is **automatic**, which enables the forwarding of native IPv6 traffic to a relay router when a public IPv4 address is assigned to any interface.
- The default resolution interval is 1440 minutes (once each day).

[↑ Top of page](#)

## show relay

Displays the relay router configuration for the 6to4 service.

### Syntax

**show relay**

[↑ Top of page](#)

## set routing

Sets both the state of routing and the inclusion of site-local address prefixes in Router Advertisements that are sent by the 6to4 router.

### Syntax

**set routing** [[**routing=**] {**enabled** | **disabled** | **automatic** | **default**}] [[**sitelocals=**] {**enabled** | **disabled** | **default**}]

### Parameters

**[[ routing=] { enabled| disabled| automatic| default}]**

Specifies whether the state of routing on a 6to4 router is enabled, disabled, automatically enabled if Internet Connection Sharing (ICS) is enabled, or set to its default value.

**[[ sitelocals=] { enabled| disabled| default}]**

Specifies whether the advertising of site-local address prefixes, in addition to 6to4 address prefixes, is enabled, disabled, or set to its default value.

### Remarks

- The default setting for the **routing=** parameter is **automatic**, which enables routing on private interfaces when ICS is used.
- The default setting for the **sitelocals=** parameter is **enabled**, which enables the advertising of site-local prefixes when site-local addresses are configured on private interfaces.

### Note

- Internet Connection Sharing and Network Bridge are not included in Windows Server 2003, Web Edition; Windows Server 2003, Datacenter Edition; and the Itanium-based versions of the original release of the Windows Server 2003 operating systems.

[↑ Top of page](#)

## show routing

Displays the routing configuration of the 6to4 service.

### Syntax

**show routing**

[↑ Top of page](#)

## set state

Configures the state of the 6to4 service.

### Syntax

```
set state [[state=] { enabled | disabled | default}] [[undoonstop=] { enabled | disabled | default}] [[6over4=] { enabled | disabled | default}]
```

### Parameters

```
[[ state=] { enabled| disabled| default}]
```

Specifies whether the state of the 6to4 service is enabled, disabled, or set to its default value.

```
[[ undoonstop=] { enabled| disabled| default}]
```

Specifies whether the reversal of all automatic configuration that has been performed by the 6to4 service occurs when the service stops is enabled, disabled, or set to its default value.

### Remarks

- The default setting for the **state=** parameter is **enabled**, which enables the 6to4 service.
- The default setting for the **undoonstop=** parameter is **enabled**, which reverses all automatic configuration performed by the 6to4 service when the service is stopped.

[↑ Top of page](#)

### show state

Displays the state of the 6to4 service.

### Syntax

```
show state
```

[↑ Top of page](#)

### reset

Resets the 6to4 service.

### Syntax

```
reset
```

[↑ Top of page](#)

## Netsh interface ipv6 isatap

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is an address assignment and tunneling mechanism for communication between IPv6/IPv4 nodes within an IPv4 site. It is described in the Internet draft titled "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)" (draft-ietf-ngtrans-isatap-00.txt). You can use the following commands to configure the ISATAP router.

- [set router](#)
- [show router](#)

### set router

Specifies the Intra-Site Automatic Tunneling Address Protocol (ISATAP) router information, including router name, state, and resolution interval.

### Syntax

```
set router [[name=]{ String | default}] [[state=]{ Enabled | Disabled | Default}] [[interval]=Integer]
```

## Parameters

**[[ name=]{ *String*| default}]**

Specifies whether the router is named with a string. If **default** is specified, the system reverts to using the default name.

**[[ state=]{ Enabled| Disabled| Default}]**

Specifies whether the ISATAP router relays packets between subnets.

**[[ interval]= *Integer*]**

Specifies the router resolution interval, in minutes. The default interval is 1440 (24 hours).

## Examples

The following example command sets the router name to isatap, enables the router, and sets the resolution interval to 120 minutes:

```
set router isatap enabled 120
```

[↑ Top of page](#)

## show router

Displays configuration information for the ISATAP router.

## Syntax

```
show router
```

## Remarks

This command displays the router name, the relay state, and the resolution interval.

[↑ Top of page](#)

## Formatting legend

Format	Meaning
<i>Italic</i>	Information that the user must supply
<b>Bold</b>	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
<b>Courier font</b>	Code or program output

## Related Links

- [Netsh commands for Interface Portproxy](#)
- [Netsh overview](#)
- [Command shell overview](#)
- [IP Version 6](#)

---

[Manage Your Profile](#)

© 2008 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

**Microsoft**