

Using Netsh

By Archiveddocs

Archived: 2026-04-05 13:27:53 UTC

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a computer that is currently running. **Netsh** also provides a scripting feature that allows you to run a group of commands in batch mode against a specified computer. **Netsh** can also save a configuration script in a text file for archival purposes or to help you configure other servers.

Netsh contexts

Netsh interacts with other operating system components using dynamic-link library (DLL) files. Each Netsh helper DLL provides an extensive set of features called a context, which is a group of commands specific to a networking component. These contexts extend the functionality of **netsh** by providing configuration and monitoring support for one or more services, utilities, or protocols. For example, Dhcpmon.dll provides **netsh** the context and set of commands necessary to configure and manage DHCP servers.

To run a **netsh** command, you must start **netsh** from the Cmd.exe prompt and change to the context that contains the command you want to use. The contexts that are available to you depend on which networking components you have installed. For example, if you type **dhcp** at the Netsh command prompt, you change to the DHCP context, but if you do not have DHCP installed the following message appears:

The following command was not found: dhcp.

For more information about Netsh contexts, see the following topics:

- [Netsh commands for AAAA](#)
- [Netsh commands for DHCP](#)
- [Netsh diagnostic \(diag\) commands](#)
- [Netsh commands for Interface IP](#)
- [Netsh commands for RAS](#)
- [Netsh commands for Routing](#)
- [Netsh commands for WINS](#)

Using multiple contexts

A context can exist within a context. For example, within the Routing context, you can change to the IP and IPX subcontexts.

To display a list of commands and subcontexts that you can use within a context, at the **netsh** prompt, type the context name, and then type either **/?** or **help**. For example, to display a list of subcontexts and commands that you can use in the Routing context, at the **netsh** prompt (that is, netsh>), type either of the following:

routing /?

routing help

To perform tasks in another context without changing from your current context, type the context path of the command you want to use at the **netsh** prompt. For example, to add the Local Area Connection interface in the IGMP context without changing to the IGMP context, at the **netsh** prompt, type:

routing ip igmp add interface "Local Area Connection" startupqueryinterval=21

Running Netsh commands from the Cmd.exe command prompt

When you run Netsh from the Cmd.exe command prompt, **netsh** uses the following syntax. 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 Windows XP.

To view the command syntax, click the following command:

```
netsh
```

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a currently running computer. Used without parameters, **netsh** opens the Netsh.exe command prompt (that is, netsh>).

Syntax

```
netsh [-a AliasFile] [-c Context] [-r RemoteComputer] [{NetshCommand|-f ScriptFile}]
```

Parameters

-a : Returns you to the **netsh** prompt after running *AliasFile*.

AliasFile : Specifies the name of the text file that contains one or more **netsh** commands.

-c : Changes to the specified **netsh** context.

Context : Specifies the **netsh** context. The following table lists the available **netsh** contexts.

Context	Description
AAAA	Shows and sets the configuration of the authentication, authorization, accounting, and auditing (AAAA) database used by the Internet Authentication Service (IAS) and the Routing and Remote Access service.
DHCP	Administers DHCP servers and provides an equivalent alternative to console-based management.
Diag	Administers and troubleshoots operating system and network service parameters.
Interface	Configures the TCP/IP protocol (including addresses, default gateways, DNS servers, and WINS servers) and displays configuration and statistical information.
RAS	Administers remote access servers.
Routing	Administers Routing servers.
WINS	Administers WINS servers.

-r : Configures a remote computer.

RemoteComputer : Specifies the remote computer to configure.

NetshCommand : Specifies the **netsh** command that you want to run.

-f : Exits Netsh.exe after running the script.

ScriptFile : Specifies the script that you want to run.

/? : Displays help at the command prompt.

- Using **-r**

If you specify **-r** followed by another command, **netsh** executes the command on the remote computer and then returns to the Cmd.exe command prompt. If you specify **-r** without another command, **netsh** opens in remote mode. The process is similar to using **set machine** at the Netsh command prompt. When you use **-r**, you set the target computer for the current instance of **netsh** only. After you exit and reenter **netsh**, the target computer is reset as the local computer. You can run **netsh** commands on a remote computer by

specifying a computer name stored in WINS, a UNC name, an internet name to be resolved by the DNS server, or a numerical IP address.

Running Netsh commands from the Netsh.exe command prompt

Netsh uses the following standard commands in all contexts that you can run from a Netsh.exe command prompt (that is, netsh>). 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 Windows XP.

To view the command syntax, click a command:

..

Moves to the context that is one level up.

Syntax

..

Parameters

/? : Displays help at the command prompt.

abort

Discards any changes made in offline mode. **Abort** has no effect in online mode.

Syntax

abort

Parameters

/? : Displays help at the command prompt.

add helper

Installs the helper DLL in **netsh**.

Syntax

add helper *DLLName*

Parameters

DLLName : Required. Specifies the name of the helper DLL you want to install.

/? : Displays help at the command prompt.

alias

Adds an alias that consists of a user-defined character string, which **netsh** treats as equivalent to another character string. Used without parameters, **alias** displays all available aliases.

Syntax

alias [*AliasName*] [*string1* [*string2* ...]]

Parameters

alias [*AliasName*] : Displays the specified alias.

alias [*AliasName*] [*string1* [*string2* ...]] : Sets *AliasName* to the specified string(s).

/? : Displays help at the command prompt.

Examples

The following **netsh** sample script sets two **netsh** aliases, Shaddr and Shp, and then leaves the Netsh command prompt in the Interface IP context:

```
alias shaddr show interface ip addr
alias shp show helpers
interface ip
```

If you type **shaddr** at the Netsh command prompt, Netsh.exe interprets this as the command **show interface ip addr**. If you type **shp** at the Netsh command prompt, Netsh.exe interprets this as the command **show helpers**.

bye

Exits Netsh.exe.

Syntax

bye

Parameters

/? : Displays help at the command prompt.

commit

Commits any changes made in the offline mode to the router. **Commit** has no effect in online mode.

Syntax

commit

Parameters

/? : Displays help at the command prompt.

delete helper

Removes the helper DLL from **netsh**.

Syntax

delete helper *DLLName*

Parameters

DLLName : Required. Specifies the name of the helper DLL you want to uninstall.

/? : Displays help at the command prompt.

dump

Creates a script that contains the current configuration. If you save this script to a file, you can use the file to restore configuration settings that have been changed. Used without parameters, **dump** displays all of the **netsh** context configurations.

Syntax

dump [*FileName*]

Parameters

[**FileName**] : Specifies the name of the file to which you want to redirect output.

/? : Displays help at the command prompt.

exec

Loads a script file and runs commands from it.

Syntax

exec *ScriptFile*

Parameters

ScriptFile : Required. Specifies the name of the script that you want to load and run.

- The *ScriptFile* can run on one or more computers.

exit

Exits Netsh.exe.

Syntax

exit

Parameters

/? : Displays help at the command prompt.

help

Displays help.

Syntax

{/?|?|help|h}

Parameters

none

offline

Sets the current mode to offline.

Syntax

offline

Parameters

/? : Displays help at the command prompt.

- Changes that you make in this mode are saved, but you need to run the **commit** or **online** command to set the changes in the router.
- When you switch from offline mode to online mode, changes that you made in offline mode are reflected in the configuration that is currently running.
- Changes that you make in online mode are immediately reflected in the configuration that is currently running.

online

Sets the current mode to online.

Syntax

online

Parameters

/? : Displays help at the command prompt.

- Changes that you make in online mode are immediately reflected in the configuration that is currently running.
- When you switch from offline mode to online mode, changes that you made in offline mode are reflected in the configuration that is currently running.

popd

Restores a context from the stack.

Syntax

popd

Parameters

/? : Displays help at the command prompt.

- Used in conjunction with **pushd**, **popd** enables you to change the context, run the command in the new context, and then resume the prior context.

Examples

The following sample script changes a context from the root context to the **interface ip** context, adds a static IP route, and then returns to the root context:

```
netsh>
```

pushd

```
netsh>
```

interface ip

```
netsh interface ip>
```

set address local static 10.0.0.9 255.0.0.0 10.0.0.1 1

```
netsh interface ip>
```

popd

```
netsh>
```

pushd

Saves the current context on a first-in-last-out (FILO) stack.

Syntax

pushd

Parameters

/? : Displays help at the command prompt.

- Used in conjunction with **popd**, **pushd** enables you to change the context, run the command in the new context, and then resume the prior context.

quit

Exits Netsh.exe.

Syntax

quit

Parameters

/? : Displays help at the command prompt.

set file

Copies the Command Prompt window output to a file.

Syntax

set file {open *FileName*|append *FileName*|close}

Parameters

open *FileName* : Sends the Command Prompt window output to the specified file.

append *FileName* : Appends the Command Prompt window output to the specified existing file.

close : Stops sending output and closes a file.

/? : Displays help at the command prompt.

- Using *FileName*

If the specified *FileName* does not currently exist, **netsh** creates a new file with that name. If the specified *FileName* does currently exist, **netsh** overwrites the existing data.

Examples

To create a new log file called Session.log and copy all succeeding **netsh** input and output to Session.log, type:

```
set file open c:\session.log
```

```
set machine
```

Sets the current computer on which to perform configuration tasks. Used without parameters, **set machine** sets the local computer.

Syntax

```
set machine [[ComputerName**=**]string]
```

Parameters

ComputerName : Specifies the name of the computer on which to perform configuration tasks.

string : Specifies the location of the remote computer.

/? : Displays help at the command prompt.

- Performing configuration tasks on multiple computers

You can run commands on multiple computers from a single script. You can use **set machine** in a script to specify a destination computer (for example, ComputerA), and then run the commands that follow **set machine** on this computer (that is, ComputerA). You can then use **set machine** to specify another destination computer (for example, ComputerB), and then run commands on this computer (that is, ComputerB).

```
set mode
```

Sets the current mode to online or offline.

Syntax

set mode {**online**|**offline**}

Parameters

online : Sets the current mode to online.

offline : Sets the current mode to offline.

/? : Displays help at the command prompt.

show

Displays alias, helper, and mode information.

Syntax

show {**alias**|**helper**|**mode**}

Parameters

alias : Lists all defined aliases.

helper : Lists all top-level helpers.

mode : Displays the current mode.

/? : Displays help at the command prompt.

unalias

Deletes the specified alias.

Syntax

unalias *AliasName*

Parameters

AliasName : Required. Specifies the name of the alias.

/? : Displays help at the command prompt.

Formatting legend

Format	Meaning
<i>Italic</i>	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

Source: <https://technet.microsoft.com/library/bb490939.aspx>