

## Netsh commands for DHCP

Updated: January 21, 2005

### Netsh commands for DHCP

The Netsh commands for DHCP offer a command-line tool that helps with the administration of DHCP servers and provides an equivalent alternative to console-based management. This can be useful in the following situations:

- When managing DHCP servers in wide area networks (WANs), commands can be used in interactive mode at the Netsh command prompt to better manage across slow-speed network links.
- When managing a large number of DHCP servers, commands can be used in batch mode at the Netsh command prompt to help script and automate recurring administrative tasks that need to be performed for all DHCP servers.

You can run these commands from the Windows Server 2003 family command prompt or from the command prompt for the Netsh DHCP context. For these commands to work at the Windows Server 2003 family command prompt, you must type **netsh dhcp** 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.

For more information about how to use the Netsh commands that are provided for DHCP, see [Use DHCP Command-line Tools](#) [<http://technet2.microsoft.com/WindowsServer/en/library/Oafbe4e4-f31d-47ae-81c8-c48aed1a34e11033.mspx>] .

For an example of how to use Netsh commands for DHCP in batch files and scripts, see [Netsh DHCP example](#) [<http://technet2.microsoft.com/WindowsServer/en/library/09e89260-0759-4d6a-8fca-cf98b34cb1cd1033.mspx>] .

For more information about Netsh commands, see [The Netsh Command-Line Utility](#) [<http://technet2.microsoft.com/WindowsServer/en/library/fd1e2fbe-15a6-413b-b712-28afb312c92f1033.mspx>] .

For more information about **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 DHCP](#)
- [Netsh DHCP server](#)
- [Netsh DHCP server scope](#)
- [Netsh DHCP server mscope](#)

### Netsh DHCP

The following commands are available at the **dhcp>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

- [add server](#)
- [delete server](#)
- [server](#)
- [show server](#)

#### add server

Adds a DHCP server to the list of authorized servers in Active Directory.

#### Syntax

**addserver** *ServerDNSServerIP*

## Parameters

### *ServerDNS*

Required. Specifies the DHCP server to add. Identifies the server by DNS domain name.

### *ServerIP*

Required. Specifies the DHCP server to add. Identifies the server by IP address.

## Examples

In the following example, this command adds a server with the DNS name `dhcprsv1.example.microsoft.com` and IP address `10.2.2.2` to the list of authorized servers in Active Directory.

```
add server dhcprsv1.example.microsoft.com 10.2.2.2
```

[↑ Top of page](#)

## delete server

Deletes a DHCP server from the list of authorized servers in Active Directory.

## Syntax

```
deleteserver ServerDNS ServerIP
```

## Parameters

### *ServerDNS*

Required. Specifies the DHCP server to delete. Identifies the server by DNS domain name.

### *ServerIP*

Required. Specifies the DHCP server to delete. Identifies the server by IP address.

## Examples

In the following example, this command deletes the server with the DNS name `dhcprsv1.example.microsoft.com` and IP address `10.2.2.2` from the list of authorized servers in Active Directory.

```
delete server dhcprsv1.example.microsoft.com 10.2.2.2
```

[↑ Top of page](#)

## server

Shifts the current Netsh DHCP command-line context to a different DHCP server. Used without parameters, **server** shifts the current command-line context to the local computer.

## Syntax

```
server [{ \\ServerNetBIOSName | \\ServerFQDN | ServerIP}]
```

## Parameters

{ \\ServerNetBIOSName | \\ServerFQDN | ServerIP }

Specifies the DHCP server to which you want to shift the current command-line context. Identifies the server by NetBIOS name, fully qualified domain name (FQDN), or IP address.

## Examples

In the first example, this command changes the Netsh DHCP command-line context to the DHCP server with the NetBIOS name `\\DHCP-SRV1`.

In the second example, this command changes the Netsh DHCP command-line context to the DHCP server with IP address `10.0.0.1`.

In the third example, this command changes the Netsh DHCP command-line context to the DHCP server with the FQDN `\\example.microsoft.com`.

server \\DHCP-SRV1

server 10.0.0.1

server \\example.microsoft.com

[↑ Top of page](#)

## show server

Displays a list of authorized servers in Active Directory.

### Syntax

showserver

### Parameters

none

[↑ Top of page](#)

[↑ Top of page](#)

## Netsh DHCP server

The following commands are available at the **dhcp server>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

- [add class](#)
- [add mscope](#)
- [add optiondef](#)
- [add scope](#)
- [delete class](#)
- [delete dnscredentials](#)
- [delete mscope](#)
- [delete optiondef](#)
- [delete optionvalue](#)
- [delete scope](#)
- [delete superscope](#)
- [dump](#)
- [export](#)
- [import](#)
- [initiate auth](#)
- [mscope](#)
- [scope](#)
- [set auditlog](#)
- [set databasebackupinterval](#)
- [set databasebackuppath](#)
-

[set databasecleanupinterval](#)

- [set databaseloggingflag](#)
- [set databasename](#)
- [set databasepath](#)
- [set databaserestoreflag](#)
- [set detectconflictretry](#)
- [set dnsconfig](#)
- [set dnscredentials](#)
- [set optionvalue](#)
- [set server](#)
- [set userclass](#)
- [set vendorclass](#)
- [show all](#)
- [show auditlog](#)
- [show bindings](#)
- [show class](#)
- [show detectconflictretry](#)
- [show dnsconfig](#)
- [show dnscredentials](#)
- [show mibinfo](#)
- [show mscope](#)
- [show optiondef](#)
- [show optionvalue](#)
- [show scope](#)
- [show server](#)
- [show dbproperties](#)
- [show serverstatus](#)
- [show userclass](#)
- [show vendorclass](#)
- [show version](#)

## add class

Adds a class to the specified DHCP server.

### Syntax

```
add class ClassName [ClassComment] [Data] [[IsVendor=]{0 | 1}] [[IsBinary=]{B | b}]
```

### Parameters

#### ***ClassName***

Required. Specifies the name of the class to create.

#### ***ClassComment***

Specifies the comment to associate with the class.

#### ***Data***

Specifies class data in ASCII or binary format depending on the **IsBinary** parameter. If unspecified, it is assumed that the format of the data is ASCII.

**[ IsVendor=]{0 | 1}**

Indicates whether the class to add is a user class or a vendor class: **0**-User class (default), **1**-Vendor class.

**[ IsBinary=]{B | b}**

Indicates if the class data is in binary format. If unspecified, it is assumed that the format of the data is ASCII.

### Examples

In the first example, this command creates a vendor class named `MyClass`, with the comment `TestClass`, and the class data `TestDataForMyClass` in ASCII format.

In the second example, this command creates a user class named `MyUserClass` with binary data of `4345335532` (ASCII-CE3U2).

The third example is an alternate command syntax for the second example.

```
add class MyClass TestClass TestDataForMyClass 1
```

```
add class MyUserClass TestClass1 4345335532 B
```

```
add class MyUserClass TestClass1 4345335532 0 b
```

[↑ Top of page](#)

### add mscope

Adds a multicast scope to the specified DHCP server.

#### Syntax

```
add mscope MscopeName [MscopeComment] [TTL]
```

#### Parameters

##### *MscopeName*

Required. Specifies the name of the multicast scope to add.

##### *MscopeComment*

Specifies a description of the current multicast scope.

##### *TTL*

Specifies the Time-To-Live value for the multicast scope. The valid range for this field is 1 to 255, with a default of 32.

### Examples

In the following example, this command creates a multicast scope with the name `My MultiCast Scope`, a comment of `MyTestMulticastScope`, and a multicast TTL value of 40.

```
add mscope "My MultiCast Scope" MyTestMulticastScope 40
```

[↑ Top of page](#)

### add optiondef

Adds a new option type definition to the specified DHCP server.

#### Syntax

```
add optiondef OptCode OptName { BYTE | WORD | DWORD | STRING | IPADDRESS } [[IsArray=]{0 | 1}]  
[vendor= VendorClass] [comment= OptComment] [DefValue]
```

#### Parameters

##### *OptCode*

Required. Specifies the option type to define. Identifies the option type with a unique option code. For the value of the code, choose a number between 0 and 255.

### ***OptName***

Required. Specifies the name of the option type.

**{BYTE | WORD | DWORD | STRING | IPADDRESS}**

Required. Indicates the data type of the option.

**[IsArray={0 | 1}]**

Indicates whether the data type is arrayed or non-arrayed: **0**-Unary or non-arrayed (default), **1**-Arrayed.

**vendor= *VendorClass***

Specifies the vendor class for the option. If unspecified, the default vendor class, the DHCP standard options class, is used.

**comment= *OptComment***

Specifies a comment to add to the new option type definition. If unspecified, the default value is NULL.

### ***DefValue***

Specifies a default value to use for this option type, if one is not specified otherwise.

### **Remarks**

- **IsArray** must be **0** for STRING type options.

### **Examples**

In the first example, this command defines a new option type named ExtensionsPath with code 18, a STRING data type, and joins it to the DHCP standard options class, assigning it a default value of c:\Temp.

In the second and third examples, this command defines a new option type named TestOption with code 231 and an IPADDRESS data type. In the second example, the option is defined to provide only a single IP address. IsArray and DefValue are not necessary because defaults are assumed. In the third example, DefValue, and IsArray are required to create an array containing the IP address values 10.1.1.1, 10.1.1.2, and 10.1.1.3.

```
add optiondef 18 ExtensionsPath STRING 0 c:\Temp
```

```
add optiondef 231 TestOption IPADDRESS
```

```
add optiondef 231 TestOption IPADDRESS 1 vendor=MyVendorClass 10.1.1.1 10.1.1.2 10.1.1.3
```

[↑ Top of page](#)

### **add scope**

Adds a scope to the specified DHCP server.

### **Syntax**

```
add scope ScopeAddressSubnetMaskScopeName [ScopeComment]
```

### **Parameters**

#### ***ScopeAddress***

Required. Specifies the scope to add. Identifies the scope by IP network number.

#### ***SubnetMask***

Required. Specifies the subnet mask for the scope to add.

#### ***ScopeName***

Required. Specifies an identifying name for the scope.

#### ***ScopeComment***

Specifies a comment for the scope.

### **Examples**

In the following example, this command adds a scope for the DHCP server with scope address 10.2.2.0, subnet mask 255.255.255.0, a scope name of MyScope, and a scope comment of MyComment.

```
add scope 10.2.2.0 255.255.255.0 MyScope MyComment
```

[↑ Top of page](#)

## **delete class**

Deletes a class from the specified DHCP server.

### **Syntax**

```
delete class Name
```

### **Parameters**

#### ***Name***

Required. Specifies the user or vendor class to delete.

### **Examples**

In the following example, this command deletes the class named MyClass.

```
delete class MyClass
```

[↑ Top of page](#)

## **delete dnscredentials**

Deletes the credentials used for all DNS dynamic updates.

### **Syntax**

```
delete dnscredentials dhcpforceflag
```

### **Parameters**

none

[↑ Top of page](#)

## **delete mscope**

Deletes a multicast scope from the specified DHCP server.

### **Syntax**

```
delete mscopeMscopeName
```

### **Parameters**

#### ***MscopeName***

Required. Specifies the multicast scope to delete. *MscopeName* is case-sensitive.

### **Examples**

In the following example, this command deletes the multicast scope MyMulticastScope from the currently specified DHCP server.

```
delete mscope MyMulticastScope
```

[↑ Top of page](#)

## **delete optiondef**

Deletes a defined option type from the specified DHCP server.

### **Syntax**

```
delete optiondefOptCode [[vendor=]VendorName]
```

## Parameters

### *OptCode*

Required. Specifies the unique identifier for the option type to delete.

### **vendor= *VendorName***

Specifies the vendor class name with which the option type is associated. If unspecified, the option type definition is deleted from the non-vendor specific DHCP standard options class.

## Examples

In the following example, this command deletes an option type with code 18 from the DHCP standard options class.

```
delete optiondef 18
```

[↑ Top of page](#)

## **delete optionvalue**

Deletes the current value of the specified option type for the specified DHCP server.

## Syntax

```
delete optionvalue OptCode [vendor= VendorClass] [user= UserClass]
```

## Parameters

### *OptCode*

Required. Specifies the unique identifier of the global server option with a value to delete.

### **vendor= *VendorClass***

Specifies the vendor class name with which the option is associated. If the tag is not included, the global *VendorName* that was set by **set vendorclass** is assumed. If the tag is used, but no value is specified, then no vendor class is used. This parameter is available only for computers running Windows 2000 Server.

### **user= *UserClass***

Specifies the user class name with which the option is associated. If the tag is not included, the global *UserName* that was set by **set userclass** is assumed. If the tag is used, but no value is specified, then no user class is used. This parameter is available only for computers running Windows 2000 Server.

## Examples

In the following example, this command deletes the current value of the option type with code 18.

```
delete optionvalue 18
```

[↑ Top of page](#)

## **delete scope**

Deletes a scope from the specified DHCP server.

## Syntax

```
delete scope ScopeAddress { DHCPNOFORCE | DHCPFULLFORCE }
```

## Parameters

### *ScopeAddress*

Required. Specifies the address of the scope to delete.

### **{ DHCPNOFORCE | DHCPFULLFORCE }**

Required. Indicates the force with which the scope will be deleted: **DHCPNOFORCE**-Deletes the scope if there are no active clients in the scope, **DHCPFULLFORCE**-Deletes the scope even if there are active clients in the scope.

## Examples

In the following example, this command forces the deletion of the scope 10.2.2.0 from the DHCP server and ignores all



warnings.

## delete scope 10.2.2.0 dhcpfullforce

[↑ Top of page](#)

## delete superscope

Deletes a superscope from the specified DHCP server.

### Syntax

```
delete superscope SuperscopeName
```

### Parameters

#### *SuperscopeName*

Required. Specifies the name of the superscope to delete.

### Remarks

- You can set the superscope from the scope context with the **set superscope** command.
- The superscope name is case-sensitive.

### Examples

In the following example, this command deletes the superscope named My Superscope from the specified DHCP server.

```
delete superscope "My Superscope"
```

[↑ Top of page](#)

## dump

Dumps the configuration of the local DHCP server to the command prompt window when run within the netsh environment.

### Syntax

```
dump
```

When run at the command prompt or in a batch file, output can be saved in a text file. Used without parameters, this command dumps the configuration of the local server to the command prompt window.

### Syntax

```
netsh dhcp server [IPAddress] dump > [PathAndFileName]
```

### Parameters

#### *IPAddress*

Specifies the IP address of the DHCP server for which the configuration is output.

#### *PathAndFileName*

Specifies both the location where the file is saved, and the name of the destination file to which to dump DHCP server configuration. If unspecified, the DHCP server configuration is dumped to the command prompt window.

### Remarks

- This command dumps the DHCP configuration as text output within the command prompt window. You can copy or redirect this output to a file to view it in a text editor. After file output is obtained, you can use the **exec** command to configure another DHCP server with the same configuration.
- Before performing the **dump** command at the source server, reconcile all scopes and fix any reported inconsistencies.
- The following command dumps the current configuration for the local DHCP server (the source) to a text file named Dhcpcfg.dmp. This file can be copied to another (destination) DHCP server and used to recreate the source configuration on it. You can recreate the configuration with the following command:

```
dump > dhcpcfg.dmp
```

- Before you use the **exec** command to execute a dump file at the destination server, complete the following steps:
  1. Close the DHCP console at the destination server. Do not reopen the console until after you have run the **exec** command.
  2. Delete all default option definitions and any user and vendor classes at the destination server.
- These examples demonstrate how to use the **delete optiondef** and **del class** Netsh commands to delete all default option definitions and any user and vendor classes. You can run these commands at the **dhcp server>** prompt, from within the **netsh** environment.
  - **delete optiondef 76**
  - **del class "Default BOOTP Class"**
  - **del class "Microsoft Windows 98 Options"**
- On the destination server, the **exec** command is used to load and execute the saved configuration:
 

```
netsh exec dhcpcfg.dmp
```
- After you use the **exec** command to load the dump file, you must reconcile all scopes. Use **net stop dhcpserver** to stop the DHCP Server service and **net start dhcpserver** to restart it. Once the service is restarted, DHCP database changes take effect.

## Examples

The first command, which is run within the netsh environment, dumps the configuration of the local DHCP server to the command prompt window.

The second command, which is run at the command prompt, dumps the configuration of a DHCP server with IP address 192.168.0.1 to a text file named Dhcpcfg.dmp at the location C:\Dhcp\ on the local computer.

The third command, which is run at the command prompt, dumps the configuration of a DHCP server with IP address 192.168.0.1 to a text file named Dhcpcfg.dmp on a shared network folder named \\Backup\Dhcp\.

## dump

```
netsh dhcp server 192.168.0.1 dump > C:\Dhcp\Dhcpcfg.dmp
```

```
netsh dhcp server 192.168.0.1 dump > \\Backup\Dhcp\Dhcpcfg.dmp
```

[↑ Top of page](#)

## export

Exports the DHCP service configuration to a file.

### Syntax

```
export [Path] FileName { all | ScopeList }
```

### Parameters

**[*Path*] *FileName***

Required. Specifies, by name, the file where the DHCP configuration will be stored. If the path, the file name, or both contain spaces, quotation marks must be used.

**{ **all** | *ScopeList* }**

Required. Specifies which scopes you want to export. The parameter **all** exports all scopes. The parameter *ScopeList* exports the scopes that correspond to the IP addresses you list. Each IP address in the list must be separated by spaces.

### Remarks

- This command works only on the local server.
- While the **export** command runs, the DHCP service is stopped and does not respond to DHCP clients seeking new leases or lease renewals.
- If the DHCP service has a large number of scopes or a large number of client address leases, this command can take a long time to run.

## Examples

In the first example, this command exports the complete DHCP service configuration to the file c:\Temp\Dhcpdb.

In the second example, this command exports the DHCP configuration for scopes 10.0.0.0 and 192.168.0.0 to the file c:\Temp\Dhcpdb.

In the third example, this command exports the complete DHCP service configuration to the file c:\My Folder\Dhcp Configuration. Note that both the path and file name contain spaces, so quotation marks are used.

```
export c:\Temp\Dhcpdb all
```

```
export c:\Temp\Dhcpdb 10.0.0.0 192.168.0.0
```

```
export "c:\My Folder\Dhcp Configuration" all
```

[↑ Top of page](#)

## import

Imports a DHCP service configuration from a file to the local service.

### Syntax

```
import [Path]FileName { all | ScopeList }
```

### Parameters

**[*Path*] *FileName***

Required. Specifies, by name, the file from which the DHCP configuration will be imported. If the path, the file name, or both contain spaces, quotation marks must be used.

**{ **all** | *ScopeList* }**

Required. Specifies which scopes you want to import. The parameter **all** imports all scopes represented in the file you specify. The parameter *ScopeList* imports the scopes that correspond to the IP addresses you list. Each IP address in the list must be separated by spaces.

### Remarks

- This command works only on the local server.
- While the **import** command runs, the DHCP service is stopped and does not respond to DHCP clients seeking new leases or lease renewals.
- If the DHCP service has a large number of scopes or a large number of client address leases, this command can take a long time to run.

### Examples

In the first example, this command imports the complete DHCP service configuration from the file c:\Temp\Dhcpdb.

In the second example, this command imports the DHCP configuration for scopes 10.0.0.0 and 192.168.0.0 from the file c:\Temp\Dhcpdb

In the third example, this command imports the complete DHCP service configuration from the file c:\My Folder\Dhcp Configuration. Note that both the path and file name contain spaces, so quotation marks are used.

```
import c:\Temp\Dhcpdb all
```

```
import c:\Temp\Dhcpdb 10.0.0.0 192.168.0.0
```

```
import "c:\My Folder\Dhcp Configuration" all
```

[↑ Top of page](#)

## initiate auth

Initiates authorization of the specified DHCP server in Active Directory.

### Syntax

```
initiate auth
```

## Parameters

none

[↑ Top of page](#)

## mscope

Switches the command context to a DHCP multicast scope.

## Syntax

**mscope** *MscopeName*

## Parameters

### *MscopeName*

Required. Specifies the name of the multicast scope to which to switch the command context.

## Remarks

- This command enables you to use any of the available DHCP multicast scope-level commands on the specified DHCP multicast scope. Subsequent operations will be performed on this multicast scope at the specified server.
- The name of the multicast scope is case-sensitive.

## Examples

In the following example, this command switches the command context to a multicast scope named My Multicast Scope.

**mscope "My Multicast Scope"**

[↑ Top of page](#)

## scope

Switches the command context to a DHCP scope.

## Syntax

**scope** *ScopeIPAddress*

## Parameters

### *ScopeIPAddress*

Required. Specifies the scope to which to switch the command context. Identifies the scope by network or subnet IP address.

## Remarks

- This command enables you to use any of the available DHCP scope-level commands on the specified DHCP scope. Subsequent operations will be performed on this scope at the specified server.

## Examples

In the following example, this command switches the command context to the scope with IP address 10.0.0.0.

**scope 10.0.0.0**

[↑ Top of page](#)

## set auditlog

Sets the audit log path parameters for the specified DHCP server.

## Syntax

**set auditlog***NewAuditPath*

## Parameters

### ***NewAuditPath***

Required. Specifies a local folder for the DHCP server to use when storing audit log files. A valid folder path is required, and the folder must be local to the DHCP server where the path is defined.

## Remarks

- In order for this change to take effect, you need to stop and restart the DHCP service after you run this command.

## Examples

In the following example, this command sets the DHCP server to store audit log files in the directory c:\logpath.

**set auditlog c:\logpath**

[↑ Top of page](#)

## **set databasebackupinterval**

Sets the backup interval for the specified DHCP server.

## Syntax

**set databasebackupinterval***NewInterval*

## Parameters

### ***NewInterval***

Required. Specifies the backup interval in minutes.

## Examples

In the following example, this command sets the database backup interval for the specified DHCP server to 1,440 minutes (24 hours).

**set databasebackupinterval 1440**

[↑ Top of page](#)

## **set databasebackuppath**

Sets the database backup path for the specified DHCP server.

## Syntax

**set databasebackuppath***NewBackupPath*

## Parameters

### ***NewBackupPath***

Required. Specifies a local folder for the DHCP server to use when storing audit log files. A valid folder path is required, and the folder must be local to the DHCP server where the path is defined.

## Examples

In the following example, this command sets the database backup path to c:\windows\dhcp\backup.

**set databasebackuppath c:\windows\dhcp\backup**

[↑ Top of page](#)

## **set databasecleanupinterval**

Sets the database cleanup interval of the specified DHCP server.

## Syntax

**set databasecleanupinterval***NewInterval*

## Parameters

***NewInterval***

Required. Specifies the database cleanup interval, in minutes.

## Examples

In the following example, this command sets the database cleanup interval to 10,080 minutes (every seven days).

**set databasecleanupinterval 10080**

[↑ Top of page](#)

## set databaseloggingflag

Sets or resets the database logging flag for the specified DHCP server. Used without parameters, this command resets the database logging flag.

## Syntax

**set databaseloggingflag** [{0 | 1}]

## Parameters

{0 | 1}

Indicates the database logging flag setting: **0**-Resets the flag (default), **1**-Sets the flag.

## Examples

In the following example, this command resets the database logging flag for the currently specified DHCP server.

**set databaseloggingflag 0**

[↑ Top of page](#)

## set databasename

Sets the name of the DHCP server database file for the specified DHCP server.

## Syntax

**set databasename***NewFileName*

## Parameters

***NewFileName***

Required. Specifies a new name for the DHCP server database file.

## Remarks

- In order for this change to take effect, you need to stop and restart the DHCP Server service.

## Examples

In the following example, this command sets the DHCP server database file name to newdatabase.mdb.

**set databasename newdatabase.mdb**

[↑ Top of page](#)

## set databasepath

Sets the path of the DHCP server database file for the specified DHCP server.

## Syntax

**set databasepath** *NewPath*

## Parameters

### *NewPath*

Required. Specifies the path of the DHCP database file.

## Remarks

- In order for this change to take effect, you need to stop and restart the DHCP Server service.

## Examples

In the following example, this command sets the path of the DHCP server database file to c:\windows\dhcp.

```
set databasepath c:\windows\dhcp
```

[↑ Top of page](#)

## set databaserestoreflag

Sets or resets the database restore flag for the specified DHCP server. Used without parameters, this command resets the database restore flag.

## Syntax

```
set databaserestoreflag [{0 | 1}]
```

## Parameters

{0 | 1}

Indicates the setting of the database restore flag: **0**-Resets the flag (default), **1**-Sets the flag.

## Examples

In the following example, this command sets the database restore flag for the currently specified DHCP server.

```
set databaserestoreflag 1
```

[↑ Top of page](#)

## set detectconflictretry

Sets the number of conflict detection attempts for the specified DHCP server.

## Syntax

```
set detectconflictretry RetryNumber
```

## Parameters

### *RetryNumber*

Required. Specifies the number of conflict detection attempts that the DHCP server will make before it leases a scope IP address to a client.

## Remarks

- The maximum value for *RetryNumber* is five. All values greater than five will default down to five.

## Examples

In the following example, this command sets the DHCP server to make four attempts to detect if there is a conflict for a scope IP address before the DHCP server uses this scope IP address in a new address lease.

```
set detectconflictretry 4
```

[↑ Top of page](#)

## set dnsconfig

Sets the DNS dynamic update configuration for the specified DHCP server.

### Syntax

```
set dnsconfig [Enable=]{0 | 1} [[Update=]{0 | 1}] [[Lookup=]{0 | 1}] [[NonDyn=]{0 | 1}]
```

### Parameters

**[Enable=]{0 | 1}**

Required. Indicates whether to enable or disable the dynamic updates of DNS client information: **0**-Disables dynamic updates of DNS client information, **1**-Enables dynamic updates of DNS client information.

**[Update=]{0 | 1}**

Indicates the type of update lookups: **0**-Updates according to client request, **1**-Updates forward and reverse name lookups.

**[Lookup=]{0 | 1}**

Indicates whether to enable or disable forward lookups for clients when leases expire: **0**-Disables forward lookups, **1**-Enables forward lookups.

**[NonDyn=]{0 | 1}**

Indicates whether to update DNS information for clients that do not support dynamic updates: **0**-Disables updates for clients that do not support dynamic updates, **1**-Enables these updates.

### Remarks

- This command is only supported for DHCP servers running Windows 2000 Server and members of the Windows Server™ 2003 family.
- The configured Preferred DNS server, in the TCP/IP properties of the DHCP server, will be updated with DHCP client records when dynamic updates are enabled. If the Preferred DNS server is not available on the network, the Alternate DNS server will be updated.

### Examples

In the following example, this command enables the dynamic update of DNS information by the DHCP server and configures updates to always be made for both forward and reverse lookups. This example also disables the option to perform forward lookups when leases expire and enables updates for non-dynamic clients.

```
set dnsconfig 1 1 0 1
```

[↑ Top of page](#)

## set dnscredentials

Sets DNS dynamic update credentials for the specified DHCP server.

### Syntax

```
set dnscredentials UserName Domain Password
```

### Parameters

***UserName***

Required. The user name associated with a valid domain account.

***Domain***

Required. The domain in which the user account is located.

***Password***

Required. The password for the user account.

### Examples

In the first example, this command sets the DNS dynamic update credentials to the values specified.

In the second example, this command sets the value of *UserName* to User1 and the value of *Domain* to Domain1, but the password is not specified. Instead an asterisk (\*) is used. When an asterisk (\*) is used in place of a valid password, the user is prompted for a password. Without the correct password for the specified *UserName*, the credentials cannot be changed.



**set dnscredentials User1 Domain1 Password1**

**set dnscredentials User1 Domain1 \***

[↑ Top of page](#)

## set optionvalue

Sets a DHCP server option value that will be applied for all scopes defined at the specified DHCP server.

### Syntax

```
set optionvalueOptCode {BYTE | WORD | DWORD | STRING | IPADDRESS} [[user=]UserName] [[vendor=]VendorName] [OptionValue]
```

### Parameters

#### *OptCode*

Required. Specifies the unique identifier for the option type with the value to set.

{**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**}

Required. Specifies the data type for the option type with the value to set.

[**user=**]*UserName*

Sets the applicable user class name to which this command will apply. If unspecified, the default user class is assumed.

[**vendor=**]*VendorName*

Sets the applicable vendor class name to which this command will apply. If unspecified, the DHCP standard options class is assumed.

[*OptionValue*]

Specifies the new server default value for the option type identified by *OptCode*. The value must be of the associated data type.

### Remarks

- Vendor and user classes are supported only for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class, use either **set userclass** or **set vendorclass** commands.

### Examples

In the first example, this command sets the value of option code 003 to list two router IP addresses, 10.1.1.1 and 10.1.1.2.

In the second example, this command further qualifies the first example and sets the specified value to apply only for those scope clients that identify themselves as members of the vendor class Vendor1. In order for this example to work, this class must already be defined at the server, with the specified option type defined for its use.

```
set optionvalue 003 IPADDRESS 10.1.1.1 10.1.1.2
```

```
set optionvalue 003 IPADDRESS vendor=Vendor1 10.1.1.1 10.1.1.2
```

[↑ Top of page](#)

## set server

Changes the Netsh command-line context to the specified server. Used without parameters, the local server is assumed.

### Syntax

```
set server[{ ServerIP | \\ServerDNS}]
```

### Parameters

{ *ServerIP* | **\\ServerDNS**}

Specifies the IP address or the DNS name of the specified DHCP server.

### Remarks

- When this command is used, and the Netsh command-line context is changed to another server, all subsequent

command operations are performed on the specified server. To perform command operations on another server, this command must be executed again, with the IP address or DNS name of the new server provided as the parameter.

### Examples

In the following example, this command switches the DHCP server command-line context to the server at IP address 10.1.1.1

```
set server 10.1.1.1
```

[↑ Top of page](#)

### set userclass

Sets the name for the current user class. Used without parameters, the current user class is reset to the default user class

#### Syntax

```
set userclass [UserClass]
```

#### Parameters

##### *UserClass*

Specifies the name of the user class for the currently specified DHCP server.

#### Remarks

- This command is available only for use with DHCP servers running Windows 2000 Server.

### Examples

In the first example, this command sets the user class name to MyUserClass.

In the second example, this command resets the current user class to the default user class.

```
set userclass MyUserClass
```

```
set userclass
```

[↑ Top of page](#)

### set vendorclass

Sets the name for the current vendor class. Used without parameters, the current vendor class is reset to the default class, DHCP standard options.

#### Syntax

```
set vendorclass [VendorClass]
```

#### Parameters

##### *VendorClass*

Specifies the name of the vendor class for the current server.

#### Remarks

- This command is available only for use with DHCP servers running Windows 2000 Server.

### Examples

In the first example, this command sets the vendor class name to MyVendorClass.

In the second example, this command resets the current vendor class to the default class, DHCP standard options.

```
set vendorclass MyVendorClass
```

```
set vendorclass
```

[↑ Top of page](#)

## **show all**

Displays all status and configuration information for the specified DHCP server.

### **Syntax**

**show all**

### **Parameters**

**none**

### **Remarks**

- This command lists all current server status and configuration details, including the stored management information base (MIB) information, for the currently specified DHCP server.

[↑ Top of page](#)

## **show auditlog**

Displays all audit log information for the specified DHCP server.

### **Syntax**

**show auditlog**

### **Parameters**

**none**

[↑ Top of page](#)

## **show bindings**

Displays bindings information for the specified DHCP server.

### **Syntax**

**show bindings**

### **Parameters**

none

### **Remarks**

- This command displays bindings information for the current DHCP server, indicating which active network connections that are configured with static IP addresses are enabled or disabled for use in servicing DHCP clients on your network.
- The DHCP service automatically disables network connections that obtain their IP address configuration dynamically from service bindings. This happens by default.

[↑ Top of page](#)

## **show class**

Enumerates and displays all class information for the specified DHCP server. This includes both user and vendor class information.

### **Syntax**

**show class**

### **Parameters**

none

[↑ Top of page](#)

### **show detectconflictretry**

Displays the configured number of conflict detection attempts for the specified DHCP server.

#### **Syntax**

**show detectconflictretry**

#### **Parameters**

none

#### **Remarks**

- This command displays the current number of ping retries that are used to attempt the detection of address conflicts for the scope IP addresses that are distributed by the server.

[↑ Top of page](#)

### **show dnsconfig**

Displays the DNS dynamic update configuration for the specified DHCP server.

#### **Syntax**

**show dnsconfig**

#### **Parameters**

none

#### **Remarks**

- If DNS configuration has not been enabled or set, you can use the DNS console or the **set dnsconfig** command to configure these settings.

[↑ Top of page](#)

### **show dnscredentials**

Displays the current DNS dynamic update credentials.

#### **Syntax**

**show dnscredentials**

#### **Parameters**

none

[↑ Top of page](#)

### **show mibinfo**

Displays management information base (MIB) information for the specified DHCP server.

## Syntax

**show mibinfo**

## Parameters

none

[↑ Top of page](#)

## show mscope

Displays all information about multicast scopes for the specified DHCP server.

## Syntax

**show mscope**

## Parameters

**none**

[↑ Top of page](#)

## show optiondef

Displays all defined and available options types for use at the specified DHCP server. Used without parameters, all options are displayed.

## Syntax

**show optiondef** [**vendor=***VendorName*]

## Parameters

**vendor=** *VendorName*

Indicates which defined and available option types to display. If unspecified, the default for *VendorName* is the default vendor class that is currently set for the DHCP server.

## Remarks

- The default vendor class can be reset with the **set vendorclass** command.
- *VendorName* is applicable only for DHCP servers running Windows 2000 Server.

## Examples

In the first example, this command displays all defined option types for the current vendor class.

In the second example, this command displays the defined option types that are available for use with the specified class Vendor1.

**show optiondef**

**show optiondef vendor=Vendor1**

[↑ Top of page](#)

## show optionvalue

Displays all of the available option values that are currently set for the specified DHCP server. Used without parameters, all option values are displayed, including those that are used for both user and vendor classes.

## Syntax

**show optionvalue**[{ **user=***UserName* | **vendor=***VendorName*}]

## Parameters

**user=** *UserName*

Specifies the default values that are set for the option types that are available for use with the currently set user class.

**vendor=** *VendorName*

Specifies the default values that are set for the option types that are available for use with the currently set vendor class.

## Remarks

- **User=** *UserName* and **vendor=** *VendorName* are only available for use with DHCP servers running Windows 2000 Server.
- If you provide values for *UserName* or *VendorName*, the respective tags (**user=** or **vendor=**) are required. If the tags are not included, the default that is assumed is the current user class, previously set with the **set userclass** command.

## Examples

In the following example, this command displays all of the available option types that have values set at the currently specified DHCP server.

**show optionvalue**

[↑ Top of page](#)

## show scope

Displays information about the scopes for the specified DHCP server.

## Syntax

**show scope**

## Parameters

**none**

[↑ Top of page](#)

## show server

Displays information about the specified DHCP server, including the fully qualified domain name and IP address of the server.

## Syntax

**show server**

## Parameters

**none**

[↑ Top of page](#)

## show dbproperties

Displays information about server database configuration for the specified DHCP server.

## Syntax

**show dbproperties**

## Parameters

**none**

## Remarks

- This command displays the following database configuration information:
  - DatabaseName
  - DatabasePath
  - DatabaseBackupPath
  - DatabaseBackupInterval
  - DatabaseLoggingFlag
  - DatabaseRestoreFlag
  - DatabaseCleanupInterval

[↑ Top of page](#)

## show serverstatus

Displays status information for the specified DHCP server.

### Syntax

**show serverstatus**

### Parameters

none

## Remarks

- This is an example of the type of status information that this command displays:

```
Server Attrib - Server Servicing Clients :TRUE
Server Attrib - Dynamic BootP Support Enabled :TRUE
Server Attrib - DHCP Server Part Of DS :TRUE
Server Attrib - DHCP Server Bindings Aware :TRUE
Server Attrib - Administrative Rights :TRUE
```

[↑ Top of page](#)

## show userclass

Displays the current user class setting at the specified DHCP server.

### Syntax

**show userclass**

### Parameters

none

## Remarks

- This is an example of the type of setting information that this command displays:

```
Current Class Name set for the Server dhcprsv1.example.microsoft.com is None.
```

[↑ Top of page](#)

## show vendorclass

Displays the current vendor class setting at the specified DHCP server.

### Syntax

**show vendorclass**

### Parameters

**none**

### Remarks

- This is an example of the type of setting information that this command displays:

```
Current Vendor Name set for the Server dhcpsrv1.example.microsoft.com is None.
```

[↑ Top of page](#)

## show version

Displays current version information for the specified DHCP server.

### Syntax

**show version**

### Parameters

**none**

### Remarks

- This is an example of the type of version information that this command displays:

```
The version of the DHCP Server 192.168.0.100 is 5.6.
```

[↑ Top of page](#)

[↑ Top of page](#)

## Netsh DHCP server scope

The following commands are available at the **dhcp server scope>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

- [add excluderange](#)
- [add iprange](#)
- [add reservedip](#)
- [delete excluderange](#)
- [delete iprange](#)
- [delete lease](#)
- [delete optionvalue](#)
- [delete reservedip](#)
- [delete reservedoptionvalue](#)
- [dump](#)



- [initiate reconcile](#)
- [set comment](#)
- [set name](#)
- [set optionvalue](#)
- [set reservedoptionvalue](#)
- [set scope](#)
- [set state](#)
- [set superscope](#)
- [show clients](#)
- [show clientsv5](#)
- [show excluderange](#)
- [show iprange](#)
- [show optionvalue](#)
- [show reservedip](#)
- [show reservedoptionvalue](#)
- [show scope](#)
- [show state](#)

## add excluderange

Adds a range of addresses to exclude from distribution in the current scope.

### Syntax

**add excluderange***StartIP**EndIP*

### Parameters

#### ***StartIP***

Required. Specifies the IP address that starts the exclusion range.

#### ***EndIP***

Required. Specifies the IP address that ends the exclusion range.

### Examples

In the following example, this command excludes the IP addresses in the range 10.2.2.10 to 10.2.2.20 from distribution in the scope.

```
add excluderange 10.2.2.10 10.2.2.20
```

[↑ Top of page](#)

## add iprange

Adds a range of IP addresses to the current scope.

### Syntax

**add iprange***StartIP**EndIP* [{ **DHCP** | **BOOTP** | **BOTH**}] [*MaxBootP*]

### Parameters

#### ***StartIP***

Required. Specifies the IP address that starts the range.

#### ***EndIP***

Required. Specifies the IP address that ends the range.

**{DHCP | BOOTP | BOTH}**

Specifies the type of client to be serviced by this scope and IP range. **DHCP** is the default.

**MaxBootP**

Specifies the maximum number of BOOTP clients.

#### Remarks

- *MaxBootP* is useful only if *ClientType* is set to support BOOTP-type clients for the scope. If an optional command-line option is used, all command-line options (mandatory as well as optional) that come before the optional command that is used are required and must appear in proper sequence.

#### Examples

In the following example, this command adds DHCP clients that are in the IP address range 10.2.2.10 to 10.2.2.20 to the distribution range in the scope.

```
add iprange 10.2.2.10 10.2.2.20
```

[↑ Top of page](#)

#### add reservedip

Reserves an IP address for use by a specified media access control (MAC) address in the current scope.

#### Syntax

```
add reservedip ReservedIP MACAddress [ClientName] [ClientComment] [{DHCP | BOOTP | BOTH}]
```

#### Parameters

**ReservedIP**

Required. Specifies the IP address to reserve.

**MACAddress**

Required. Specifies the physical hardware or MAC address to associate with the reserved IP address.

**ClientName**

Specifies a client name to associate with this reserved client entry. If unspecified, a client name is not associated with this reserved client entry.

**ClientComment**

Specifies a client comment to associate with this reserved client entry. If unspecified, a client comment is not associated with this reserved client entry.

**{DHCP | BOOTP | BOTH}**

Specifies the type of clients to associate with this reserved client entry. **DHCP** is the default.

#### Remarks

- If an optional command-line option is used, all command-line options (mandatory as well as optional) that come before the optional command-line option that is used are required and must appear in proper sequence.

#### Examples

In the following example, this command reserves the IP address 10.2.2.32 for use by the DHCP client that identifies its MAC address as 08-00-2b-30-36-9b when it obtains a lease in the current scope.

```
add reservedip 10.2.2.32 08002b30369B
```

[↑ Top of page](#)

#### delete excluderange

Deletes a range of previously excluded IP addresses from the current scope.

#### Syntax

**delete excluderange** *StartIP* *EndIP*

#### Parameters

##### ***StartIP***

Required. Specifies the IP address that starts the exclusion range.

##### ***EndIP***

Required. Specifies the IP address that ends the exclusion range.

#### Examples

In the following example, this command deletes the exclusion range that starts with an IP address of 10.2.1.5 and ends with an IP address of 10.2.1.10 from the current scope.

**delete excluderange 10.2.1.5 10.2.1.10**

[↑ Top of page](#)

#### **delete iprange**

Deletes a range of IP addresses from the current scope.

#### Syntax

**delete iprange** *StartIP* *EndIP*

#### Parameters

##### ***StartIP***

Required. Specifies the IP address that starts the range to delete.

##### ***EndIP***

Required. Specifies the IP address that ends the range to delete.

#### Examples

In the following example, this command deletes IP addresses in the range 10.2.1.5 to 10.2.1.10 from the current scope.

**delete iprange 10.2.1.5 10.2.1.10**

[↑ Top of page](#)

#### **delete lease**

Deletes an IP address lease from the current scope.

#### Syntax

**delete lease** [*LeaseIP* | *\\HostName* | **AIIBadAddresses** | **AIIRasServerAddresses**]

#### Parameters

**{ *LeaseIP* | *\\HostName* | **AIIBadAddresses** | **AIIRasServerAddresses** }**

Required. Specifies the IP address or the host name associated with the leased address. If **AIIBadAddresses** is specified, all records marked as BAD\_ADDRESS are removed. If **AIIRasServerAddresses** is specified, all leases obtained by all Routing and Remote Access Service servers are removed.

#### Examples

In the first example, this command deletes the IP address lease 10.2.1.32 from the current scope.

In the second example, this command deletes all IP address leases marked as BAD\_ADDRESS from the current scope.

**delete lease 10.2.1.32**

**delete lease AIIBadAddresses**

## delete optionvalue

Removes or clears the currently set scope option value from the current scope.

### Syntax

```
delete optionvalue OptCode[user= UserName] [vendor= VendorName]
```

### Parameters

#### *OptCode*

Required. Specifies the unique identifier of the option whose value is to delete.

#### **user**= *UserName*

Specifies the user class from which to delete the currently set option value. If the tag is provided, but the value unspecified, the current default user class is assumed.

#### **vendor**= *VendorName*

Specifies the vendor class from which to delete the currently set option value. If the tag is provided, but the value unspecified, the current default vendor class is assumed.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- If you specify a user class, a vendor class, or both, this command deletes the set option value from only the scope clients that are identified as members of the specified class or classes.
- To modify the current defaults for an unspecified class that is used with this command, use either the **set userclass** or **set vendorclass** commands.

### Examples

In the first example, this command deletes the currently set option value for option code 18 from the current scope.

In the second example, this command deletes the currently set option value for option code 18 from the vendor class MyVendorClass.

```
delete optionvalue 18
```

```
delete optionvalue 18 vendor="MyVendorClass"
```

## delete reservedip

Deletes a reservation for an IP address in the current scope.

### Syntax

```
delete reservedip ReservedIPMACAddress
```

### Parameters

#### *ReservedIP*

Required. Specifies the IP address reservation to delete from the current scope.

#### *MACAddress*

Required. Specifies the media access control (MAC) or physical hardware address string for which the IP address was reserved.

### Examples

In the following example, this command removes the IP address reservation 10.2.1.32 for the MAC address 08002B30369B from the current scope.

```
delete reservedip 10.2.1.32 08002B30369B
```

## delete reservedoptionvalue

Deletes an option value that is currently assigned to a reserved client in the current scope.

### Syntax

```
delete reservedoptionvalue ReservedIPOptCode [User= UserName] [vendor= VendorName]
```

### Parameters

#### *ReservedIP*

Required. Specifies the IP address of the reserved client.

#### *OptCode*

Required. Specifies the unique code for the option type that is currently assigned to the reserved client.

#### **User=** *UserName*

Specifies the user class from which to delete the option value. If the tag is provided, but no value is specified, the current default user class is assumed.

#### **vendor=** *VendorName*

Specifies the vendor class from which to delete the option value. If the tag is provided, but no value is specified, the current vendor class is assumed.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

### Examples

In the following example, this command deletes the currently set option value for the option identified by code 18 for reserved IP address of 10.2.2.32 in the current scope.

```
delete reservedoptionvalue 10.2.2.32 18
```

## dump

Dumps the configuration of the current scope to the command prompt window when run within the netsh environment.

### Syntax

#### dump

When run at the command prompt or in a batch file, output can be saved in a text file. Used without parameters, this command dumps the configuration of the current scope to the command prompt window.

### Syntax

```
netsh dhcp server { ServerName | IPAddress } scope ScopeID dump > [PathAndFileName]
```

### Parameters

#### *ServerName* | *IPAddress*

Specifies the server name or IP address of the DHCP server for which the scope configuration is output.

#### *ScopeID*

Required. Specifies the IP address of the scope for which the configuration is output.

#### *PathAndFileName*

Specifies both the location where the file is saved, and the name of the destination file to which to dump the scope configuration. If unspecified, the scope configuration is dumped to the command prompt window.

## Examples

The first command, which is run within the netsh environment, dumps the configuration of the current scope to the command prompt window.

The second command, which is run from the command prompt, dumps the configuration of the local server scope 192.168.1.0 to a file named Scopecfg.dmp on the shared network folder \\Backup\Dhcp\.

## dump

```
netsh dhcp server scope 192.168.1.0 dump > \\Backup\Dhcp\Scopecfg.dmp
```

[↑ Top of page](#)

## initiate reconcile

Checks and reconciles the current scope. Used without parameters, this command verifies the scopes and check for inconsistencies but does not fix any inconsistencies that it finds in the database.

## Syntax

```
initiate reconcile [fix]
```

## Parameters

### [fix]

Indicates that the command will fix, if possible, any inconsistencies that it finds in the database.

## Examples

In the first example, this command verifies the scopes and checks for inconsistencies.

In the second example, this command verifies the scopes, checks for inconsistencies, and fixes any inconsistencies that are found.

## initiate reconcile

```
initiate reconcile fix
```

[↑ Top of page](#)

## set comment

Sets the comment for the current scope. Used without parameters, **set comment** deletes the current comment.

## Syntax

```
set comment [NewComment]
```

## Parameters

### [NewComment]

Specifies a new or modified comment string for the scope.

## Examples

In the first example, this command sets a comment string that has no spaces.

In the second example, this command sets a comment string that includes spaces.

```
set comment NewCommentNoSpaces
```

```
set comment "New Comment With Spaces"
```

[↑ Top of page](#)

## set name

Sets the name of the current scope.

### Syntax

**set name** *NewName*

### Parameters

#### ***NewName***

Required. Specifies the new name of the scope.

### Examples

In the first example, this command sets a name that has no spaces.

In the second example, this command sets a name that includes spaces.

**set name NewNameNoSpaces**

**set name "New Name With Spaces"**

[↑ Top of page](#)

## set optionvalue

Sets an option value for the current scope.

### Syntax

**set optionvalue** *OptCode*{**BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS**} [**user=***UserName*] [**vendor=***VendorName*]  
*OptionValue*

### Parameters

#### ***OptCode***

Required. Specifies the code for the option type whose value is to be set.

**{BYTE | WORD | DWORD | STRING | IPADDRESS}**

Required. Specifies the data type for the option type whose value is to be set.

**user=***UserName*

Specifies the user class. If the tag is provided, but the value is unspecified, the current default user class is assumed.

**vendor=***VendorName*

Specifies the vendor class. If the tag is provided, but the value is unspecified, the current default vendor class is assumed.

#### ***OptionValue***

Required. Specifies the assigned value for the option type that is specified in *OptCode*. If the option type supports an array that contains more than a single numeric or IP address value, provide the additional values, in the order that you prefer them, at the end of the command, with each value separated by a space.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

### Examples

In the first example, this command sets the value of option code 003 to list two router IP addresses (10.1.1.1, 10.1.1.2).

In the second example, this command sets the value that is specified in the first example to apply only to those scope clients that identify themselves as members of the vendor class Vendor1, a class that was previously defined at the server with this specified option type defined for its use.

**set optionvalue 003 IPADDRESS 10.1.1.1 10.1.1.2**

**set optionvalue 003 IPADDRESS vendor=Vendor1 10.1.1.1 10.1.1.2**

[↑ Top of page](#)

## set reservedoptionvalue

Sets the value of an option for a reservation IP address in the current scope.

### Syntax

```
set reservedoptionvalue ReservedIPOptCode { BYTE | WORD | DWORD | STRING | IPADDRESS } [user=UserName]  
[vendor=VendorName] OptValue
```

### Parameters

#### *ReservedIP*

Required. Specifies the reserved IP address for which the option value is to be set.

#### *OptCode*

Required. Specifies the code for the option type whose value is to be set.

{ **BYTE** | **WORD** | **DWORD** | **STRING** | **IPADDRESS** }

Required. Specifies the data type for the option type whose value is to be set.

**user=***UserName*

Specifies either the current default user class or the class specified as *UserName*. If the tag is provided, but no value is specified, the current default user class is assumed.

**vendor=***VendorName*

Specifies either the current default vendor class or the class specified as *VendorName*. If the tag is provided, but no value is specified, the current default vendor class is assumed.

#### *OptValue*

Required. Specifies the assigned value for the option type specified in *OptCode*. If the option type supports an array that contains more than a single numeric or IP address value, provide the additional values, in the order that you prefer them, at the end of the command with each value separated by a space.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

### Examples

In the following example, this command sets the value of the router option (code 003) for the reserved client IP address of 10.1.1.50 in the current scope to set IP addresses of 10.1.1.1 and 10.1.1.2 for its configured routers (default gateways).

```
set reservedoptionvalue 10.1.1.50 003 IPADDRESS 10.1.1.1 10.1.1.2
```

[↑ Top of page](#)

## set scope

Sets the scope for use in subsequent operations.

### Syntax

```
set scope ScopeAddress
```

### Parameters

#### *ScopeAddress*

Required. Specifies the IP address of the scope to use in subsequent command operations.

### Examples

In the following example, this command sets the current scope to 10.2.2.0 for subsequent operations.

```
set scope 10.2.2.0
```

[↑ Top of page](#)



## set state

Sets or resets the state of the current scope to either the active or inactive state. Used without parameters, this command activates the scope.

### Syntax

```
set state [{0 | 1 | 2 | 3}]
```

### Parameters

{0 | 1 | 2 | 3}

Indicates the state of the scope: **0**-Deactivates the scope, **1**-Activates the scope (default), **2**-Deactivates the scope and marks the scope as "Switched," **3**-Activates the scope and marks the scope as "Switched."

### Remarks

- **2** and **3** are typically used for switched networks or networks where multiple logical networks are hosted on a single physical network.

### Examples

In the first example, this command activates a scope.

In the second example, this command deactivates a scope.

```
set state 1
```

```
set state 0
```

[↑ Top of page](#)

## set superscope

Sets the superscope to use in subsequent operations.

### Syntax

```
set superscope SuperscopeName {0 | 1}
```

### Parameters

*SuperscopeName*

Required. Specifies the name of the superscope to include the current scope. The *SuperscopeName* is case-sensitive.

{0 | 1}

Required. Indicates whether to set the state of the superscope to active or inactive: **0**-Deactivates the superscope, **1**-Activates the superscope.

### Examples

In the following example, this command adds the current scope to the superscope MySuperScope and activates the superscope.

```
set superscope MySuperScope 1
```

[↑ Top of page](#)

## show clients

Displays all of the available version 4 clients for the current scope. Used without parameters, this command displays the following information for each client: IP address, subnet mask, unique ID, lease expiration, and type.

### Syntax

```
show clients[{0 | 1}]
```

### Parameters

{0 | 1}

Indicates the detail level of the output: **0**-Shows the following information for each client: IP address, subnet mask, unique ID, lease expiration, and data type (default), **1**-Shows all of the information that **0** provides and also displays the fully qualified domain name of each client.

#### Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

[↑ Top of page](#)

### show clientsv5

Displays all of the available version 5 clients for the current scope. Used without parameters, this command displays the following information for each client: IP address, subnet mask, unique ID, lease expiration, and type.

#### Syntax

**show clientsv5**{**0** | **1**}

#### Parameters

{**0** | **1**}

Indicates the detail level of the output: **0**-Shows the following information for each client: IP address, subnet mask, unique ID, lease expires, type (default), **1**-Shows all of the information that **0** provides and also displays the fully qualified domain name of each client.

#### Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

[↑ Top of page](#)

### show excluderange

Displays all of the currently set exclusion ranges of IP addresses for the current scope.

#### Syntax

**show excluderange**

#### Parameters

none

#### Remarks

- Use other scope-level commands to add and delete exclusion ranges.

[↑ Top of page](#)

### show iprange

Displays all of the address ranges that are available for the current scope.

#### Syntax

**show iprange**

#### Parameters

none

[↑ Top of page](#)

## show optionvalue

Displays all of the option values that are set for the current scope. Used without parameters, this command assumes the current default user and vendor classes.

### Syntax

**show optionvalue**[*user=UserName*] [*vendor=VendorName*]

### Parameters

**user=UserName**

Specifies that the options that are set for the specified user class will display. If the tag is provided, but no value is specified, the current default user class is assumed.

**vendor=VendorName**

Specifies that the options that are set for the specified vendor class will display. If the tag is provided, but no value is specified, the current default vendor class is assumed.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

### Examples

In the following example, this command displays all options and values set for the current scope for the user defined class My User Class.

```
show optionvalue user="My User Class"
```

[↑ Top of page](#)

## show reservedip

Displays all of the IP addresses that are currently reserved for the current scope.

### Syntax

**show reservedip**

### Parameters

**none**

[↑ Top of page](#)

## show reservedoptionvalue

Displays all currently set option values for a reserved client IP address in the current scope.

### Syntax

**show reservedoptionvalue***ReservedIP*[*user=UserName*] [*vendor=VendorName*]

### Parameters

**ReservedIP**

Required. Specifies the IP address reservation for which currently assigned options are to display.

**user=UserName**

Specifies that the options that are set for the specified user class will display. If the tag is provided, but no value is specified, the current default user class is assumed.

**vendor=VendorName**

Specifies that the options that are set for the specified vendor class will display. If the tag is provided, but no value is

specified, the current default vendor class is assumed.

### Remarks

- Vendor and user classes are only supported for DHCP servers running Windows 2000 Server.
- To modify the current defaults for an unspecified class used with this command, use either **set userclass** or **set vendorclass**.

### Examples

In the following example, this command displays the option values set for the reserved IP address 10.2.2.100 in the current scope.

**show reservedoptionvalue 10.2.2.100**

[↑ Top of page](#)

### show scope

Displays information for the current scope.

### Syntax

**show scope**

### Parameters

**none**

[↑ Top of page](#)

### show state

Displays the state of the current scope, indicating whether it is active or inactive.

### Syntax

**show state**

### Parameters

**none**

[↑ Top of page](#)

[↑ Top of page](#)

## Netsh DHCP server mscope

The following commands can run from the **dhcp server mscope>** prompt, which is rooted within the **netsh** environment.

To view the command syntax, click a command:

- [add excluderange](#)
- [add iprange](#)
- [delete excluderange](#)
- [delete iprange](#)
- [dump](#)
-

[initiate reconcile](#)

- [set comment](#)
- [set lease](#)
- [set mscope](#)
- [set name](#)
- [set state](#)
- [set ttl](#)
- [show clients](#)
- [show excluderange](#)
- [show iprange](#)
- [show lease](#)
- [show mibinfo](#)
- [show mscope](#)
- [show state](#)
- [show ttl](#)

### **add excluderange**

Adds a range of excluded addresses to the current multicast scope.

#### **Syntax**

**add excluderange***StartIP**EndIP*

#### **Parameters**

##### ***StartIP***

Required. Specifies the IP address that starts the exclusion range.

##### ***EndIP***

Required. Specifies the IP address that ends the exclusion range.

#### **Remarks**

- The exclusion range must be a subset of the overall scope address range.

#### **Examples**

In the following example, this command adds an exclusion range that starts with 224.2.2.10 and ends with 224.2.2.20 to the distribution range of the current multicast scope.

**add excluderange 224.2.2.10 224.2.2.20**

[↑ Top of page](#)

### **add iprange**

Adds a range of IP addresses to the current multicast scope.

#### **Syntax**

**add iprange***StartIP**EndIP*

#### **Parameters**

##### ***StartIP***

Required. Specifies the IP address that starts the range.

### ***EndIP***

Required. Specifies the IP address that ends the range.

### **Remarks**

- The range must be within the valid range of multicast IP addresses (from 224.0.0.0 to 239.255.255.255).

### **Examples**

In the following example, this command adds the IP address range 224.2.2.10 to 224.2.2.20 to the distribution range in the multicast scope.

```
add iprange 224.2.2.10 224.2.2.20
```

[↑ Top of page](#)

### **delete excluderange**

Deletes an exclusion range of previously excluded IP addresses in the current multicast scope.

### **Syntax**

```
delete excluderangeStartIPEndIP
```

### **Parameters**

#### ***StartIP***

Required. Specifies the IP address that starts the exclusion range.

#### ***EndIP***

Required. Specifies the IP address that ends the exclusion range.

### **Examples**

In the following example, this command removes a multicast scope exclusion with a range of IP addresses that starts at 224.2.2.10 and ends with 224.2.2.20 for the multicast scope.

```
delete excluderange 224.2.2.10 224.2.2.20
```

[↑ Top of page](#)

### **delete iprange**

Deletes a range of IP addresses from the current multicast scope.

### **Syntax**

```
delete iprangeStartIPEndIP
```

### **Parameters**

#### ***StartIP***

Required. Specifies the IP address that starts the range to delete.

#### ***EndIP***

Required. Specifies the IP address that ends the range to delete.

### **Examples**

In the following example, this command deletes the IP address range 224.2.2.10 to 224.2.2.20 from the overall range of the full multicast scope.

```
delete iprange 224.2.2.10 224.2.2.20
```

[↑ Top of page](#)

### **dump**

Dumps the configuration of the current multicast scope to the command prompt window when run within the netsh environment.

## Syntax

### dump

When run at the command prompt or in a batch file, output can be saved in a text file. Used without parameters, this command dumps the configuration of the current multicast scope to the command prompt window.

## Syntax

```
netsh dhcp server [ServerName | IPAddress] mscope MscopeID dump > [PathAndFileName]
```

## Parameters

### *ServerName* | *IPAddress*

Specifies the server name or IP address of the DHCP server for which the scope configuration is output.

### *MscopeID*

Required. Specifies the name of the multicast scope for which the configuration is output.

### *PathAndFileName*

Specifies both the location where the file is saved, and the name of the destination file to which to dump the multicast scope configuration. If unspecified, scope configuration is dumped to the command prompt window.

## Examples

The first command, which is run within the netsh environment, dumps the configuration of the current multicast scope to the command prompt window.

The second command, which is run from the command prompt, dumps the configuration of the local server multicast scope TestMscope to the file Mscopecfg.dmp on the shared network folder \\Backup\Dhcp\.

### dump

```
netsh dhcp server mscope TestMscope dump > \\Backup\Dhcp\Mscopecfg.dmp
```

[↑ Top of page](#)

## initiate reconcile

Checks and reconciles the current multicast scope.

## Syntax

### initiate reconcile

## Parameters

### none

## Remarks

- This command checks the integrity of the current multicast scope by comparing the current contents of the server database with a mirrored copy of the same information in the Windows registry. If inconsistencies are detected in the database, they are repaired based on the information that is duplicated in the registry. Repair is always attempted for any inconsistency that is found.

[↑ Top of page](#)

## set comment

Sets the comment for the current multicast scope.

## Syntax

**set comment** *NewComment*

### Parameters

#### ***NewComment***

Required. Specifies the new or revised comment for the multicast scope.

### Examples

In the first example, this command modifies the multicast scope comment with a new comment that contains no spaces.

In the second example, this command modifies the multicast scope comment with a new comment that contains spaces.

**set comment NewCommentNoSpaces**

**set comment "New Comment With Spaces"**

[↑ Top of page](#)

### set lease

Sets the lease duration for the multicast scope IP addresses.

### Syntax

**set lease** *Time*

### Parameters

#### ***Time***

Required. Specifies the lease duration for clients of the multicast scope. Specifying **-1** sets the duration of the IP address lease to an unlimited or infinite time.

### Examples

In the following example, this command sets the lease duration for the clients of the current multicast scope to 691200 seconds (eight days).

**set lease 691200**

[↑ Top of page](#)

### set mscope

Sets the multicast scope to use in subsequent operations.

### Syntax

**set mscope** *NewName*

### Parameters

#### ***NewName***

Required. Specifies the name of the multicast scope to which the command context is changed. *NewName* is case-sensitive.

### Remarks

- This command changes the context of the **netsh dhcp server mscope>** prompt from one multicast scope to another.
- This command does not rename the current multicast scope. To change the name of the current multicast scope, use the command **set name**.

### Examples

In the first example, this command changes the command context to a multicast scope named MyMulticastScope. Note that the multicast scope name contains no spaces.

In the second example, this command changes the command context to a multicast scope named My Multicast Scope. Note that the multicast scope name contains spaces.



**set mscope MyMulticastScope**

**set mscope "My Multicast Scope"**

[↑ Top of page](#)

## **set name**

Changes the name of the current multicast scope.

### **Syntax**

**set name***NewName*

### **Parameters**

*NewName*

Required. Specifies a new name for the current multicast scope.

### **Examples**

In the first example, this command sets a new name that contains no spaces for the current multicast scope.

In the second example, this command sets a new name that contains spaces for the current multicast scope.

**set name NewNameNoSpaces**

**set name "New Name With Spaces"**

[↑ Top of page](#)

## **set state**

Sets or resets the state of the current multicast scope to either an active or inactive state.

### **Syntax**

**set state** {0 | 1}

### **Parameters**

{0 | 1}

Required. Sets the state of the current multicast scope: **0**-Deactivates the current multicast scope, **1**-Activates the current multicast scope .

### **Examples**

In the first example, this command activates the current multicast scope.

In the second example, this command deactivates the current multicast scope.

**set state 1**

**set state 0**

[↑ Top of page](#)

## **set ttl**

Sets the Time-To-Live (TTL) value for the current multicast scope.

### **Syntax**

**set ttl***TTL*

### **Parameters**

*TTL*

Required. Specifies the Time-to-Live (TTL) value. The valid range for this value is a number from 1 to 255.

### Examples

In the following example, this command sets the TTL value for the current multicast scope to 32.

```
set ttl 32
```

[↑ Top of page](#)

### show clients

Displays all available clients for the current multicast scope.

#### Syntax

```
show clients
```

#### Parameters

none

#### Remarks

- To view the output of this command effectively, increase the width of the command prompt window to at least 95 characters.

[↑ Top of page](#)

### show excluderange

Displays all currently excluded ranges of IP addresses for the current multicast scope.

#### Syntax

```
show excluderange
```

#### Parameters

none

#### Remarks

- If no exclusion ranges have been previously defined for the scope, this command outputs an empty list.

[↑ Top of page](#)

### show iprange

Displays all available IP address ranges for the current multicast scope.

#### Syntax

```
show iprange
```

#### Parameters

none

[↑ Top of page](#)

### show lease

Displays the current lease duration settings for the current multicast scope.

### Syntax

**show lease**

### Parameters

**none**

[↑ Top of page](#)

### show mibinfo

Displays management information base (MIB) information for the current multicast scope.

### Syntax

**show mibinfo**

### Parameters

**none**

[↑ Top of page](#)

### show mscope

Displays information for the current multicast scope.

### Syntax

**show mscope**

### Parameters

**none**

[↑ Top of page](#)

### show state

Displays the state of the current multicast scope.

### Syntax

**show state**

### Parameters

**none**

[↑ Top of page](#)

### show ttl

Displays the Time-To-Live (TTL) value for the current multicast scope.

### Syntax

show ttl

## Parameters

none

[↑ Top of page](#)

[↑ 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 AAAA](#)
- [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](#)