

dhcpcd command (DHCP Server)



Successfully pass the free certification exam at IW Academy and become an Infinet Certified Engineer.

[To the certification exam](#)

- Description
 - DHCP Client
- Parameters
 - Address scope
 - Network interfaces
 - Virtual interfaces
 - General parameters
 - Configuration options
 - Address Time
- Examples

Description

The "*dhcpcd*" command is used to manage DHCP server. Parameters of this command are entered using console with the "*dhcpcd*" prefix for WANFlex command interpreter.

Parameters are not case-sensitive and allow arbitrary abbreviation up to one letter if command as a whole can not be perceived by the DHCP server ambiguously. E.g., "*dhcpcd show scope **" command can be shorted to "*dhcpcd s s **", and "*dhcpcd show config*" command as "*dhcpcd sh c*". In above command list parameters are put into <>. If parameter value contains spaces, this value must be put into quotes.



NOTE

DHCP server cannot be started simultaneously with DHCP relay, if the device configuration contains the "*dhcpr start*" command, the server start settings will not be applied.



NOTE

The commands which change DHCP configuration (including "*stop*" and "*start*" commands) can be executed only by administrator with "*super-user*" rights. Other commands can be executed by any user.

Syntax (without prefix):

```

add dscope <SCOPE_NAME> <INTERFACE|*> <START_IP> <END_IP>
add scope <SCOPE_NAME> <INTERFACE|*> <START_IP> <END_IP>
add virtual interface <VIFNAME> <GATEWAY> <GWIFNAME|*>
clear
delete option <OPTION_NAME>
delete scope <SCOPE_NAME>
delete useroption <NAME>
delete virtual interface <VIFNAME>
help options
interface <INTERFACE> delete option <OPTION_NAME>
interface <INTERFACE> option <OPTION_NAME> <OPTION_VALUE>
interface <INTERFACE> reservation <CLIENT_ID> delete option <OPTION_NAME>
interface <INTERFACE> reservation <CLIENT_ID> option <OPTION_NAME> <OPTION_VALUE>
interface <INTERFACE|*> show boundhistory
interface <INTERFACE|*> show client <CLIENT_ID|*>
lock interface <INTERFACE>
notrace
option <OPTION_NAME> <OPTION_VALUE>
scope <SCOPE_NAME> add classid <CLIENT_CLASS_ID>
scope <SCOPE_NAME> add exclude <START_IP> <END_IP>
scope <SCOPE_NAME> add reservation <CLIENT_ID> <CLIENT_IP>
scope <SCOPE_NAME> delete classid <CLIENT_CLASS_ID>
scope <SCOPE_NAME> delete exclude <START_IP>
scope <SCOPE_NAME> delete option <OPTION_NAME>
scope <SCOPE_NAME> delete reservation <CLIENT_ID>
scope <SCOPE_NAME> interface <INTERFACE|*>
scope <SCOPE_NAME> option <OPTION_NAME> <OPTION_VALUE>
scope <SCOPE_NAME> reservation <CLIENT_ID> delete option <OPTION_NAME>
scope <SCOPE_NAME> reservation <CLIENT_ID> option <OPTION_NAME> <OPTION_VALUE>
scope <SCOPE_NAME> set range <START_IP> <END_IP>
scope <SCOPE_NAME|*> show declinehistory
show config
show interface <INTERFACE|*>
show options
show scope <NAME|*>
show unleases <SUBSTR|*>
show version
show xml
start
stop
trace
unlock interface <INTERFACE>
useroption <NAME> code <CODE> type <TYPE>
virtual interface <VIFNAME> add subnet <IP_ADDRESS> <SUBNET_MASK>
virtual interface <VIFNAME> change <GATEWAY> <GWIFNAME|*>
virtual interface <VIFNAME> delete subnet <IP_ADDRESS> <SUBNET_MASK>

```

**WARNING**

The DHCP server executes commands only after its start.

```
dhcpd start
```



DHCP Client

DHCP protocol is used for (workstations and servers) TCP network hosts connection parameters dynamic configuration. UDP protocol is used as a transport protocol for DHCP. Host which requests data for its network connection configuration (IP-address, subnet mask, default gateway etc) is called DHCP client. IP-address is a basic configuration parameter. After client's start it sends a DHCP request over the network so it could get a lease of IP-address and other network parameters. For its identification in its request a client may use client identifier. In general case, client identifier is a binary set of bytes which is unique within a physical network segment to which a client is connected. If client does not provide an identifier, the server will accept client's MAC-address for network interface. Thus, in DHCP server a client is identified by its identifier and network interface from which server accepts client's requests (client's interface). Client's identifier ("*<CLIENT_ID>*" parameter in commands) is represented as ID:<identifier> or 01:<MAC-address of network adapter> as it is shown at the example below.

ID:01:00:04:35:22:88:1D


In its requests to the server, a client may indicate its class (class identifier). Class identifier is a string which defines one of client's properties which is common for a set of clients. For example, it can be client operating system's name. E.g. DHCP clients which work under OS Windows XP send "MSFT 5.0" as a class identifier, Infinet Wireless IP-phones – "IW_IP_PHONE". Client's class can be used by server administrator for automatic clients' grouping in IP-address scopes in order to conveniently assign them specific configuration parameters (options). Scope is a range of IP-addresses within which a server can assign addresses to its clients. Scopes are located in a configuration database of a server and are identified by names configured by server administrator when this scope was created.

Parameters

Parameters	Description
Address scope	
add scope <SCOPE_NAME> <INTERFACE *> <START_IP> <END_IP>	<p>Allows to create a scope, even if a scope cannot be attached to some interface, it is not deleted.</p> <ul style="list-style-type: none"> "SCOPE_NAME" – scope name. It is not case-sensitive and must be unique. If scope name contains spaces, server will automatically substitute them with "underscore" sign (_). "INTERFACE" – name of the network interface with which this scope will be attached (allowed interface). If "*" is specified as interference, that means that this scope can be attached to all suitable network interfaces. Suitable network interface is an interface which contains a subnet of IP-addresses (aliases) that includes starting and ending IP-addresses of the scope. "START_IP" / "END_IP" – starting and ending IP-addresses of the scope correspondingly. When attaching to network interface, it is checked if a range of this scope does not intersect (and is not included) within another scope that might be attached to this interface. When IP-addresses are assigned to clients, only those scopes can be used which are attached to the same network interface through which requests from clients are accepted.
scope <SCOPE_NAME> set range <START_IP> <END_IP>	<p>Changes a range of IP-addresses of existing scope.</p> <ul style="list-style-type: none"> "SCOPE_NAME" – scope name which range need to be changed. "START_IP" and "END_IP" – new starting and ending IP-addresses of a scope. <div>  WARNING When executing command, exceptions (set by the "scope exclude" parameter) which were created before range changing and which stop satisfying conditions described above, will be deleted automatically. </div>
scope <SCOPE_NAME> interface <INTERFACE *>	<p>Changes an interface for the scope.</p> <ul style="list-style-type: none"> "SCOPE_NAME" – scope name which interface need to be changed. "INTERFACE" – name of the network interface to which a scope is attached to. <div>  NOTE If a system does not have an interface with specified name or a system cannot attach this scope to specified interface, the scope will be immediately detached. This feature can be used for temporary shutdown of one of the scopes. </div>
scope <SCOPE_NAME> add classid <CLIENT_CLASS_ID>	<p>Scope of IP-addresses has clients class filter. If a client in its request submits its class, a server is able to give an IP-address only from those scopes which are connected to interface through which requests from clients are accepted. and which have client's class specified in their class filter. Class filter is a set of client vendor class id from which it is allowed to give a lease for IP-addresses from the scope. In order to create a class filter for a scope, one should add one or more client vendor class id. This parameter is used to add a client vendor class id to the scope.</p> <ul style="list-style-type: none"> "SCOPE_NAME" – name of the scope to which client vendor class id is added ("CLIENT_CLASS_ID"). "CLIENT_CLASS_ID" – a set of characters of variable length (up to 255 characters). If this parameter contains spaces it should be specified in quotes. This "<CLIENT_CLASS_ID>" is compared to what client submits when requests for IP-address lease. If client submitted a class which does not present in any of scope's filters or a client did not submit any class name, only scopes with no class filters can be used for IP-address lease.

scope <SCOPE_NAME> delete classid <CLIENT_CLASS_ID>	Deletes a class from the scope's filter.
scope <SCOPE_NAME> add exclude <START_IP> <END_IP>	<p>Sets up exceptions into scope range of addresses. Exceptions are range of addresses which belong to the scope but are not given to DHCP server clients.</p> <ul style="list-style-type: none"> • "SCOPE_NAME" – scope name to which we add exceptions. • "START_IP" and "END_IP" – starting and ending addresses of an exception. Exception's range should not intersect (or belong) with any of previous exceptions assigned to this scope. Exception's range should belong to the scope.
scope <SCOPE_NAME> delete exclude <START_IP>	Deletes exception from the scope.
scope <SCOPE_NAME> add reservation <CLIENT_ID> <CLIENT_IP>	<p>Creates a reservation for the scope. The scope reservation is using to reserve an IP-address for a specific client.</p> <ul style="list-style-type: none"> • "SCOPE_NAME" – name of the scope to which reservation is added. • "CLIENT_ID" – client identifier. • "CLIENT_IP" – IP-address which will be given to this client. Scope reservations are saved in configuration database of the server and are identified by scope name and client's identifier.
scope <SCOPE_NAME> delete reservation <CLIENT_ID>	Deletes reservation from the scope. If a client acquired its IP-address, after reservation is deleted a server will hold a lease of this address to this client if a client does violate scope's rules (exceptions and class filters).
scope <SCOPE_NAME> option <OPTION_NAME> <OPTION_VALUE>	<p>Adds a set of options for the scope.</p> <ul style="list-style-type: none"> • "SCOPE_NAME" – name of the scope to which options are added. • "OPTION_NAME" and "OPTION_VALUE" – the description of these parameters is in paragraph "Configuration options".
scope <SCOPE_NAME> delete option <OPTION_NAME>	Deletes options for the scope.
scope <SCOPE_NAME> reservation < CLIENT_ID> option <OPTION_NAME> <OPTION_VALUE>	<p>Adds a set of options for the reservation in the scope.</p> <ul style="list-style-type: none"> • "SCOPE_NAME" – scope name for which reservation an option value need to be defined. • "CLIENT_ID" – reservation client identifier. If this option with the same name was defined, the value will be changed to the one specified in this command. • "OPTION_NAME" and "OPTION_VALUE" – the description of these parameters is in paragraph "Configuration options".

scope <SCOPE_NAME> reservation <CLIENT_ID> delete option <OPTION_NAME>	Deletes options for the reservation in the scope.
show scope <NAME/*>	Displays the scope status. In the paragraph " Address Time " you can find decoding IP-address values for this option. <ul style="list-style-type: none"> "NAME" – scope name for which status need to be displayed. If "*" is specified as a name of scope, status for all scopes will be displayed.
scope <SCOPE_NAME/*> show declinehistory	Displays history of client's refusals to accept offered IP-address. When the client responds to the DHCP server with refuse, the server adds this information in the " <i>declinehistory</i> " to avoid re-offer of this IP-address to this client.
delete scope <SCOPE_NAME>	Deletes the scope.
Network interfaces	
show interface <INTERFACE/*>	Displays network interfaces information. After its start, the server automatically detects all network interfaces which are suitable for serving DHCP clients. Suitable interface is an interface connected to a multiple-access network with broadcast support (including VLAN support). In server database each interface is identified by its name which was assigned to it while registration in WANFlex OS core. <ul style="list-style-type: none"> "INTERFACE" – network interface name which information is required. If "*" is specified instead of interface name, all interfaces information is printed. Command output is a structured list.
lock interface <INTERFACE>	Locks one or several interfaces – in this case they cannot be used by DHCP server.
unlock interface <INTERFACE>	Unlock the interface.
interface <INTERFACE> option <OPTION_NAME> <OPTION_VALUE>	Assigns a set of options for the interface. <ul style="list-style-type: none"> "INTERFACE" – the name of the interface options are applied for. "OPTION_NAME" and "OPTION_VALUE" – the description of these parameters is in paragraph "Configuration options".
interface <INTERFACE> delete option <OPTION_NAME>	Cancels the assignment of options for the interface.

interface <INTERFACE> reservation <CLIENT_ID> option <OPTION_NAME> <OPTION_VALUE>	<p>Adds options for the interface's reservation. Interfaces reservations are required for specific settings for the client no matter from what scope the client is getting its IP-address lease. Interface reservation has two differences from scope reservation. At first, does not define a fixed IP-address for the client. Thus it takes for the server to dynamically define from which scope and which IP-address is to be given to the client. Secondly, allows to change client's class. If "Class ID" option is defined for the interfaces reservation, the class will be changed for the option's value when a client from this reservation sends a request. It becomes necessary when DHCP client does not send its class.</p> <p>Creating interface reservation does not contradict with scope reservation for the same client.</p> <ul style="list-style-type: none"> "INTERFACE" – name of the interface where client's (CLIENT_ID) reservation is created. If this interface did not have a reservation for this client, this command will automatically create this reservation and will add it the options set. "CLIENT_ID" – reservation client identifier. If this option with the same name was defined, the value will be changed to the one specified in this command. "OPTION_NAME" and "OPTION_VALUE" – the description of these parameters is in paragraph "Configuration options".
interface <INTERFACE> reservation <CLIENT_ID> delete option <OPTION_NAME>	<p>Deletes options for the interface's reservation.</p> <div>  NOTE If after deletion the options set for this reservation become empty, the interface reservation will be deleted automatically. </div>
interface <INTERFACE *> show client <CLIENT_ID *>	<p>Allows to control options which were requested by the client and given to it. On the map divisions (option sets) are displayed using the following indication:</p> <ul style="list-style-type: none"> "SR" – scope reservation division; "IR" – interface reservation division; "S" – scope reservation; "I" – client's interface division; "SV" – server's division. <p>If a requested option was not defined in any of server's divisions, it is displayed as <not supplied> in the list. Moreover, the options which were requested by clients and supplied to them but which were not defined in any division (e.g. Subnet Mask) are marked as "DF".</p> <ul style="list-style-type: none"> "INTERFACE" – name of a network interface which information is requested. Instead of interface name can be specified "*", this will print information for all interfaces. "CLIENT_ID" – client's identifier, which information is requested. Instead of client's identifier it is permitted to specify "*", this will print information about all clients for the specified interface. The information is shown only for clients with given address lease from one of the scopes which is attached to the specified interface.
interface <INTERFACE *> show boundhistory	<p>Displays the IP-addresses canceled leases database.</p> <ul style="list-style-type: none"> "INTERFACE" – name of a network interface which information is requested. Instead of interface name can be specified "*", this will print information for all interfaces.

Virtual interfaces

add virtual interface <VIFNAME> <GATEWAY> <GWIFNAME *>	<p>Creates virtual interface.</p> <p>After their start, DHCP clients send broadcast request in order to get an IP-address lease. As a client at this time does not yet have an IP-address the server also uses broadcast packets to communicate with a client. It is known that broadcast packets are not routed and, thus, the dialog between DHCP server and DHCP client can occur only within one network (physical network). If DHCP server is connected to another network, the direct dialog cannot take place. However, the router which logically connects two networks with DHCP client and DHCP server can have a special software running – DHCP Relay Agent (DRA). DRA retranslates DHCP packets (including broadcast packets) from DHCP clients to DHCP server and back. Data exchange between DRA and DHCP server is performed using unicast packets only. Thus, DRA and DHCP must know each other's IP-addresses starting from their configuration stage. For this purpose DHCP server has virtual interfaces. In fact DHCP server virtual interface is a physical network interface placed in DRA. As DHCP server does not know this interface's subnets sets, these subnets should be specified while virtual interfaces configuration.</p> <ul style="list-style-type: none"> "VIFNAME" – name of the DHCP server's virtual interface. "GATEWAY" – IP-address of DRA which has a corresponding physical interface. "GWIFNAME" – name of DRA's interface that performs retranslation of DHCP packets to DHCP clients. If the "*" symbol is used as "GWIFNAME" parameter DRA is allowed to use all its interfaces to retranslate DHCP packets.
--	---

virtual interface <VIFNAME> change <GATEWAY> <GWIFNAME *>	Changes settings of virtual interface. <ul style="list-style-type: none"> • "VIFNAME" – name of the DHCP server's virtual interface. • "GATEWAY" – new IP-address of DRA. • "GWIFNAME" – new name of DRA's interface.
virtual interface <VIFNAME> add subnet <IP_ADDRESS > <SUBNET_M ASK >	In server's configuration we now can observe the virtual interface. Working with this interface is no different from other interfaces. However, before a scope is attached to it, a set of subnets should be configured. <ul style="list-style-type: none"> • "VIFNAME" – name of the DHCP server's virtual interface. • "IP_ADDRESS" – IP-address which DRA has for this subnet. • "SUBNET_MASK" – subnet mask.
virtual interface <VIFNAME> delete subnet <IP_ADDRESS > <SUBNET_M ASK>	Deletes a subnet from the virtual interface's list. When deleted a subnet to which a scope was connected, the scope will be detached automatically. This scope will stay detached until an appropriate subnet is configured for the virtual interface.
delete virtual interface <VIFNAME>	Deletes virtual interface, subnets will be deleted also, scopes will be detached.
General parameters	
help options	Shows the list of all options DHCP server can include in messages for clients.
show version	Shows version of DHCP server.
show xml	Shows version of DHCP server in XML format.
trace/notrace	Enables / disables the debug trace.
useroption <NAME> code <CODE> type <TYPE>	Besides standard DHCP Server options user can create his own. <ul style="list-style-type: none"> • "NAME" – name of option. • "CODE" – the code of the created option should be unique from the standard and previously created user-defined DHCP options. • "TYPE" – the type of the option can be of the following values: <ul style="list-style-type: none"> • "n8" – 8-bit integer. • "n16" – 16-bytes integer. • "n32" – 32-bytes integer. • "ip" – IP-address. • "text" – text string (254 characters maximum) • "ipp" – IP-address pairs (classful routes). • "clr" – classless routes in the following format: A.B.C.D/M>R.R.R.R: <ul style="list-style-type: none"> • "A.B.C.D" – network address, • "M" – network mask, • "R.R.R.R" – gateway IP-address.
delete useroption <NAME>	Deletes a user-defined option.
option <OPTION_NAME> <OPTION_VALUE>	Adds options for the whole DHCP server.

delete option <OPTION_NAME>	Deletes options for the whole DHCP server.
show unleases <SUBSTR/*>	Displays the list of unleased IP-addresses. Clients to which DHCP server failed to give an IP-address for a lease are put to a special list – unleases. The records in this list are saved for 15 minutes if a client does not repeat an attempt to get a lease. Each record in the list consists of the following fields: <ul style="list-style-type: none"> • Name of a network interface from which a client's request for a lease was received. • Client's identifier • Client's class identifier • Host name • "SUBSTR" – a substring for a partial list view. When executing a command the server will print only those records which fields contain the substring (one of the fields). Substring is case-sensitive. If "*" is specified as a substring the full list is printed.
stop	Stops DHCP server.
clear	Clears DHCP server configuration (first it should be stopped by "dhcpd stop" command).

Configuration options

Configuration options might be requested by clients from the server for more precise host configuration. These parameters are Address Time, Router, NTP Servers etc. Clients may request a different set of these parameters. The parameters are only sent when a client included them in its request and only when server knows the value of the parameter. Divisions and values of the parameters are defined while DHCP server configuration. Divisions can be defined for the following purposes:

- Scope reservation. Options values from this division will be given to the client of this reservation.
- Interface reservation. Options are sent if requested option's value is not in scope's reservation divisions.
- Scope. Option values from this division can be sent to the client which received an address lease from this scope only if the option requested by the client is not in scope's or interface's reservation division.
- Interface. Sent to the client who received a lease from one of the scopes which is attached to the interface (and the value of the requested option was not in scope's reservation, in the scope itself and in interface's reservation).
- Server. Sent to clients which received a lease from one of the scopes (if the value of the option was not in all divisions listed above). Meaning of the division – default value.

If option's value does not exist in all divisions, client does not receive anything from the server. Two exceptions are possible:

- **Address Time** – the value of this parameter is always sent to the client. If this value is not specified in all divisions, the client receives a default value of 120 (lease time – 2 minutes).
- **Subnet Mask** – the value of this parameter is always sent to the client. The value of this option is automatically determined by the server and it cannot be defined in options divisions while server configuration. The value of the subnet mask for the client always equals subnet mask of the interface to which the scope is attached (this scope gave a lease to the client).

DHCP configuration options (overall table) is available using the following link: <http://www.iana.org/assignments/bootp-dhcp-parameters>

To define a set of options, DHCP server has special commands for each division. These commands have parameters, which are inputted in a common way (for all divisions):

- "**OPTION_NAME**" – name of the option (see the link for the table above). If option name has spaces, they must be substituted with " _ " sign. Option name is not case-sensitive.
- "**OPTION_VALUE**" – value of the option. Input format depends on the purpose of the option and is divided into three categories by DHCP server:
 - **Symbolic**. A string (e.g. for Bootfile-Name option). If this option's value has spaces, the option value should be put in quotes.
 - **Binary**. One or several decimal numbers. If several numbers should be specified, they are separated by commas. Options examples: Address Time, Time Offset.
 - **IP-address**. One or several values – IP-addresses. Several IP-addresses are separated by commas.



NOTE

Not all of the options can be defined in any division. Apart from Subnet Mask (was described above), there are options which can be defined for some particular divisions. If you attempt to add one of these options to inappropriate division, the server will report an error:

```
ERR: This option cannot contain in the given division
```

Moreover, there is a set of service options which although are included into a summary table, they do not act as configuration parameters but act as service parameters. The list of service options of DHCP server looks as follows:

- Subnet Mask.
- Address Request.
- Overload.
- DHCP Msg Type.
- DHCP Server Id.
- Parameter List.
- DHCP Message.
- DHCP Max Msg Size.
- Client Id.

If you attempt to add one of these options to any division, the server will report an error:

```
ERR: This option cannot contain in the given division
```

Address Time

Any IP-address lease is limited by the time specified in Address Time option. If a client which was given a lease does not extend it within Address Time period, the server will cancel the lease. The value of this time may be defined by the client but it should not exceed its maximal value. The maximal time of a lease is set up in Address Time of one of the divisions to which this client is applied. If a server does not have this option defined, the maximal time will be set to 120 seconds. In case if a client does not request Address Time parameter, the server will give a lease for a maximal time according to the scheme described above.

A client, who received a lease, confirms it periodically. The periodicity is usually equal to the half of Address Time. As an acknowledgement to the lease prolongation the server resends configuration parameters (options). Thus, if during the lease some of the options were changed in the server (or division to which this client was applied) the client will learn it in the moment of lease prolongation.

If after lease expiration the client does not confirm it, the scope cancels the lease. If the client is not a scope reservation client, the scope will mark the IP-address of this lease as "conditionally free". On scope state output ("*dhcpcd show scope **") this state will be marked as "<OBIND>". Thus, with other addresses available for lease, the scope will not give "<OBIND>" addresses for new clients. If during 24 hours from the moment of lease expiration the client will request for a lease again, the server will give him the same IP-address.

At the same time, the scope writes down the parameters of expired lease into a special database (boundhistory).

The information about expired leases is saved in the database during 24 hours. After 24 hours the record is automatically deleted from the database, and the IP-address becomes a free address (after being "<OBIND>"). The server will use "<OBIND>" addresses for other clients if all the scopes (which suit new clients) ran out of free addresses. The server will use at first the oldest records in boundhistory. The server will also cancel an address lease after a client's corresponding request.

Examples

Create the scope "MSOFT" with "eth0" as a permitted interface. Scope will be automatically connected to subnet "192.168.177.12" of "eth0".

```
dhcpcd add scope MSOFT eth0 192.168.177.20 192.168.177.22
[eth0] <192.168.177.12> (MSOFT): 192.168.177.20-192.168.177.22   Scope attached
OK
```

Title

Create the scope "new" with permission to connect to any suitable interface. A scope was successfully created but could not find a suitable interface to be attached to, because no one interface contains a subnet that includes the range of IP-addresses of this scope.

```
dhcpd add scope new * 10.12.12.30 10.12.12.50
WRN: Scope created, but not attached.
```

Thus if a client ID:01:00:04:35:00:22:23 sends a request to the interface with attached scope PHONES, the server will definitely give this client 9.1.1.170 address. IP-address of the reservation must be within a scope range. Exceptions does not affect the reservation. Even if the scope has an exception (e.g. "dhcpd scope PHONES add exclude 9.1.1.165 9.1.1.175"), to which the reserved address belongs, the server will create such a reservation anyway.

```
dhcpd scope PHONES add reservation ID:01:00:04:35:00:22:23 9.1.1.170
OK
```

If you add a reservation for some client, but for the same client exists a reservation in another scope, new reservation will not be created and a user will see an error message.

```
dhcpd scope other add reservation ID:01:00:04:35:00:22:23 192.168.177.10
[eth0] <192.168.177.12> (OTHER): 192.168.177.10-192.168.177.19 Reservation for "ID:01:00:04:35:00:22:23"
already exists in scope PHONES with IP=9.1.1.170
ERR: Reservation's IP is out of scope's range
```

In this example after the reservation was deleted, the server cancelled a lease for "ID:01:00:05:90:02:1F:C8" client for IP-address "9.1.1.200" in "PHONES" scope because client's class does not fulfill class filter requirements in the scope. After some time, the same client obtained another IP-address from "OTHER" scope.

```
dhcpd scope phones delete
      reservation ID:01:00:05:90:02:1F:C8
OK
dhcpd show scope *
>SCOPES:
(MSOFT)      192.168.177.20 - 192.168.177.22 [eth0] ATTACHED [eth0] <192.168.177.12>/255.255.255.0
<CLIENT CLASS IDS>: "IW_BRI_GATEWAY" "MSFT 5.0"
<CLIENT> ID:01:00:C0:DF:10:AF:69 "MSFT 5.0" 'wad ' 192.168.177.20 <BOUND> since 01/01/2003 01:
01:08
<FREE RANGE> 192.168.177.21 - 192.168.177.22 =2

(NEW)        10.12.12.30 - 10.12.12.50 [*]

(OTHER)      192.168.177.10 - 192.168.177.19 [eth0] ATTACHED [eth0] <192.168.177.12>/255.255.255.0
<CLIENT> ID:01:00:05:90:02:1F:C8 "" ' ' 192.168.177.10 <BOUND> since 01/01/2003 01:
16:36
<FREE RANGE> 192.168.177.11 - 192.168.177.11 =1
<FREE RANGE> 192.168.177.13 - 192.168.177.19 =7

(PHONES)     9.1.1.151 - 9.1.1.200 [*] ATTACHED [eth0] <9.1.1.100>/255.255.255.0
<CLIENT CLASS IDS>: "IW_IP_PHONE"
<CLIENT> ID:01:00:04:35:00:22:24 "IW_IP_PHONE" 'Stas ' 9.1.1.151 <BOUND> since 01/01/2003 01:01:47
<FREE RANGE> 9.1.1.152 - 9.1.1.169 =18
<RESERV> ID:01:00:04:35:00:22:23 "IW_IP_PHONE" 'Andrew ' 9.1.1.170 <BOUND> since 01/01/2003 01:01:37
<FREE RANGE> 9.1.1.171 - 9.1.1.200 =30
<OPTION> Router 9.1.1.3
<OPTION> H323_GK_ADDRESS 195.38.45.84

OK
```

In next example "ID:01:00:04:35:00:22:23" client did not have his lease cancelled (9.1.1.170 address) because this client fulfills all scope's rules.

```
dhcpd scope phones delete
      reservation ID:01:00:04:35:00:22:23
OK
dhcpd show scope phones
>SCOPES:
(PHONES)     9.1.1.151 - 9.1.1.200 [*] ATTACHED [eth0] <9.1.1.100>/255.255.255.0
<CLIENT CLASS IDS>: "IW_IP_PHONE"
<CLIENT> ID:01:00:04:35:00:22:24 "IW_IP_PHONE" 'Stas ' 9.1.1.151 <BOUND> since 01/01/2003
01:01:47
<FREE RANGE> 9.1.1.152 - 9.1.1.169 =18
<CLIENT> ID:01:00:04:35:00:22:23 "IW_IP_PHONE" 'Andrew ' 9.1.1.170 <BOUND> since 01/01/2003
01:01:37
<FREE RANGE> 9.1.1.171 - 9.1.1.200 =30
<OPTION> Router 9.1.1.3
<OPTION> H323_GK_ADDRESS 195.38.45.84

OK
```

Lock the "eth0" interface for DHCP server. After using of the "dhcpd show interface *" command we can see three scopes attached to "eth0": "phones", "other" and "msoft". "phones" is attached to "9.1.1.100/255.255.255.0" subnet, two others - to "192.168.177.12/255.255.255.0".

```
dhcpd show interface *
>INTERFACES
[eth0] UP <SUBNET> 9.1.1.100/255.255.255.0
      <SCOPE> (PHONES) 9.1.1.151 - 9.1.1.200 <SUBNET> 192.168.177.12/255.255.255.0
      <SCOPE> (OTHER) 192.168.177.10 - 192.168.177.19
      <SCOPE> (MSOFT) 192.168.177.20 - 192.168.177.22
OK
```

Lock the "eth0" interface.

```
dhcpd lock interface eth0
[eth0] <9.1.1.100> (PHONES): 9.1.1.151-9.1.1.200   Scope detached
[eth0] <192.168.177.12> (OTHER): 192.168.177.10-192.168.177.19   Scope detached
[eth0] <192.168.177.12> (MSOFT): 192.168.177.20-192.168.177.22   Scope detached
OK
```

After locking, let us see interfaces information again. We can see that now "eth0" interface is locked and it had all his scopes detached.

```
dhcpd show interface *
>INTERFACES
[eth0] UP LOCKED
<SUBNET> 9.1.1.100/255.255.255.0
<SUBNET> 192.168.177.12/255.255.255.0
OK
```

Create a scope of addresses from which a DHCP server can give a lease to the clients to which a DRA has an access to.

```
dhcpd add scope VIRTUAL_TEST vvv1 192.168.188.20 192.168.188.50
dhcpd show interface vvv1
>VIRTUAL INTERFACES
[vvv1] 192.168.177.81:* UP
      <SUBNET> 192.168.188.1/255.255.255.0
      <SCOPE> (VIRTUAL_TEST) 192.168.188.20 - 192.168.188.50
dhcpd show scope virtual_test
>SCOPES:
(VIRTUAL_TEST) 192.168.188.20 - 192.168.188.50 [vvv1] ATTACHED [vvv1] <192.168.188.1>/255.255.255.0 <FREE
RANGE> 192.168.188.20 - 192.168.188.50      =31
```

The list of client's supplied options.

```
dhcpd interface * show client *
>INTERFACES CLIENTS
----- [eth0] -----
(IPHONES) <CLIENT> ID:01:00:04:35:00:22:24 "IW_IP_PHONE" 'Unknown node' 192.168.0.101 <BOUND> since 25/04
/2005 11:32:57
SUPPLIED OPTIONS:
#1      . . . . . DF Subnet Mask                255.255.255.0
#2      . . . . . Time Offset                    <not supplied>
#3      . . S . . . Router                      192.168.0.1
#7      . . . . . Log Server                    <not supplied>
#42     . . S . . . NTP Servers                  192.168.0.1
#230    . . S . . . H323 GK ADDRESS              192.168.0.1
#231    . IR . . . . H323 LOGIN ALIAS            IWPhone/V. Pupkin/101
#232    . . . . . H323 GK ID                    <not supplied>
```