

Chapter 23

Dynamic Host Configuration Protocol (DHCP)

Introduction	23-2
Configuring the DHCP Server	23-3
BOOTP Relay Agent	23-4
DHCP Option 82 Relay	23-6
Configuring the DHCP Client	23-6
DHC Files	23-6
Configuration Example	23-7
Troubleshooting	23-8
Command Reference	23-9
add bootp relay	23-10
add dhcp option	23-11
add dhcp policy	23-13
add dhcp range	23-19
create dhcp policy	23-20
create dhcp range	23-21
delete bootp relay	23-22
delete dhcp option	23-23
delete dhcp policy	23-24
delete dhcp range	23-28
destroy dhcp policy	23-29
destroy dhcp range	23-29
disable bootp relay	23-30
disable bootp relay option82	23-30
disable dhcp	23-31
disable dhcp debug	23-32
enable bootp relay	23-32
enable bootp relay option82	23-33
enable dhcp	23-33
enable dhcp debug	23-34
purge bootp relay	23-34
set bootp relay maxhops	23-35
set bootp relay option82	23-36
set bootp relay option82 port	23-37
set dhcp	23-38
set dhcp option	23-39
set dhcp policy	23-41
set dhcp range	23-47
show bootp relay	23-48
show bootp relay port	23-50
show dhcp	23-51
show dhcp client	23-53
show dhcp policy	23-55
show dhcp range	23-57

Introduction

This chapter describes the Dynamic Host Configuration Protocol (DHCP) support provided by the router, and how to configure the router to act as either a DHCP and BOOTP server, or a BOOTP relay agent.

BOOTP Bootstrap Protocol (BOOTP) is a UDP-based protocol that enables a booting host to dynamically configure itself without external interventions. A BOOTP server responds to requests from BOOTP clients for configuration information, such as the IP address the client should use. BOOTP is defined in RFC 951, *Bootstrap Protocol (BOOTP)*.

RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*, defines extensions to the BOOTP protocol, including the behaviour of a BOOTP Relay Agent.

DHCP DHCP is based on BOOTP, and is defined in RFC 2131. It extends the BOOTP mechanism by providing:

- a method for passing configuration information to hosts on a TCP/IP network
- automatic allocation of reusable network addresses
- other additional configuration options

When the router is configured as a DHCP server, it allocates IP addresses and other IP configuration parameters to clients (hosts), when the client requests them. This lets you configure your IP network without manually configuring every client. Note that each client must also be configured to receive its IP address automatically.

As well as addresses, a DHCP server assigns a wide range of parameters to clients, including subnet information and mask, domain and hostname, server addresses, keepalive times, MTUs, boot settings, encapsulation settings, time settings, and TCP settings.

DHCP is designed to interoperate with BOOTP clients as well as DHCP clients, without the BOOTP clients needing any change to their initialization software. DHCP messages are a subset of BOOTP messages, which allows BOOTP relay agents to relay these messages.

BOOTP relay agents BOOTP relay agents pass BOOTP messages between servers and clients. Networks where the DHCP or BOOTP server does not reside on the same IP subnet as its clients need the intermediate routers to act as BOOTP relay agents.

Note that you can configure the router to act as a BOOTP relay agent or a DHCP/BOOTP server, but not both.

Configuring the DHCP Server

DHCP is based on *DHCP policies*. Policies are predefined sets of configuration information items. Each policy defines IP configuration information for the clients that are attached to a single IP interface. Each policy has at least one IP address *range* attached to it. A range is a list of consecutively numbered IP addresses. When the DHCP server uses a policy to supply DHCP information to a client, it assigns the client an unused IP address from the policy's IP address ranges.

The router supports DHCP and its predecessor, BOOTP, but they are disabled by default.

To configure the router as a DHCP server:

1. Enable IP and give the desired interface an IP address and subnet mask. This IP address needs to be in the subnet that you wish to assign to hosts that are connected to that interface. Use the command:

```
enable ip
add ip interface
```

See [Chapter 22, Internet Protocol \(IP\)](#) for information about configuring IP interfaces. If the interface is a VLAN, you may have to create it first.

2. Create a DHCP policy by using the command:

```
create dhcp policy=name leasetime={lease-time|infinity}
[inherit=name]
```

3. Assign an IP address range to the policy. This range must be in the same subnet as the IP address that you assigned to the interface. Use the command:

```
create dhcp range=name ip=ipadd number=number policy=name
[gateway=ipadd] [probe={arp|icmp}]
```

4. Assign any other desired configuration settings to the policy by using the commands:

```
add dhcp policy [other-options]
add dhcp option [other-options]
```

The server uses that policy on that interface. Repeat this process with as many interfaces and policies as required.

For more information, see [Using options](#), below.

5. Enable the DHCP server by using the command:

```
enable dhcp
```

IP settings are assigned to hosts for specific times (the *lease time*). You can use DHCP to allocate any of the following types of addresses:

- A dynamic IP address, which is available to the host for a limited amount of time (specified as the lease time) and is then reclaimed by the server. The server can then allocate it to another device on request.

To configure this, give the policy a lease time that is less than **infinity** when you create it, using the command:

```
create dhcp policy=name leasetime=lease-time
```

- A permanent IP address, which is made available to the client on request and never reclaimed. This is referred to as "Automatic" allocation.

To configure this, give the policy a lease time of **infinity** when you create it, using the command:

```
create dhcp policy=name leasetime=infinity
```

- A manual or static IP address, which is allocated to a particular client. The client is identified by its MAC address. This lets you use DHCP to manage most of your network automatically, while having unchanging IP addresses on key devices such as servers.

To configure this, add a static entry to the IP address range, using the command:

```
add dhcp range=name address=macadd ip=ipadd
```

BOOTP requests can be satisfied by policies with leases set to **infinity**.

Using options

DHCP allows the client to receive options from the DHCP server. Options describe the network configuration, and various services that are available on the network.

There are two types of options defined using two separate commands:

- Add a standard, pre-defined option to a policy using [add dhcp policy](#).
- Add a user-defined option to a policy using [add dhcp option](#).

It is possible to add a user-defined option with the same number as an existing pre-defined option. If this situation occurs, the user-defined option takes precedence - that is, it overrides but does not eliminate the standard option.

For more information, see the [add dhcp policy command on page 23-13](#) and [add dhcp option command on page 23-11](#).

BOOTP Relay Agent

BOOTP relay agents pass BOOTP messages between servers and clients. Networks where the DHCP or BOOTP server does not reside on the same IP subnet as its clients need the routers attached to the subnet to act as BOOTP relay agents.

Note that both BOOTP and DHCP use BOOTP message, allowing BOOTP relay agents to relay all their packets.

The router's BOOTP Relay Agent relays these message types:

- BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination
- BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router

The relay agent ignores BOOTREPLY messages addressed to clients on networks not directly connected to the router. The router treats these as ordinary IP packets for forwarding.

A BOOTREQUEST message may be relayed via unicast, multicast or broadcast methods. In the last case, the message does not re-broadcast to the interface from which it was received. The relay destinations are configured independently of other broadcast forwarders' destinations (e.g. TFTP).

The hops field in a BOOTP message records the number of hops (routers) the message has been through. If the value of the hops field exceeds a predefined threshold (normally 16), the relay agent discards the message. You can set the threshold to a value from 1 to 16.

Configuring the router

To enable the BOOTP Relay Agent, use the command:

```
enable bootp relay
```

To define a relay destination, use the command:

```
add bootp relay=ipadd [interface=interface]
```

You can define more than one relay destination on the router. The **interface** parameter allows you to set different relay destinations for different interfaces. Note that:

- if an interface has one relay destination defined, the router relays BOOTP packets it receives on that interface to only the relay destination
- if an interface has multiple relay destinations defined, the router relays BOOTP packets it receives on that interface to each defined relay destination
- if an interface has no relay destinations defined, the router relays BOOTP packets it receives on that interface to any relay destination where an interface was not specified for the relay destination

To delete a relay destination, use the command:

```
delete bootp relay=ipadd [interface=interface]
```

To purge the BOOTP configuration (including the relay destination list), use the command:

```
purge bootp relay
```

This disables the BOOTP relay agent and purges all configuration data (including non-volatile storage).

When the 'hops' field in a BOOTP message exceeds a predefined threshold the BOOTP message is discarded. The default of the threshold is 4. To set the threshold, use the command:

```
set bootp relay maxhops=1..16
```

To display the current configuration of the BOOTP Relay Agent, use the command:

```
show bootp relay
```

DHCP Option 82 Relay

DHCP and BOOTP functionality includes the addition, removal and monitoring of DHCP Option 82. Option 82 is also called the Relay Agent Information option.

Option 82 is inserted by the DHCP relay agent into the DHCP options field when forwarding client-originated BOOTP/DHCP packets to a DHCP server. DHCP servers that are configured to recognise Option 82 may use the information to implement IP addresses, or other parameter assignment policies, based on the network location of the client device.

For more information about Option 82, see RFC 3046.

Option 82 can be:

- added to packets relayed from the DHCP client to DHCP server
- removed from packets relayed from DHCP server to DHCP client
- checked from sources closer to the client

Configuring the DHCP Client

An interface on the router can also be configured as a DHCP client by using the command:

```
add ip interface=interface ipaddress=dhcp [other-options...]
```

and enable remote address assignment by using the command:

```
enable ip remoteassign
```

For more information, see [“DHCP Client” on page 22-19 of Chapter 22, Internet Protocol \(IP\)](#).

DHC Files

Information about the state of the DHCP server is stored in flash memory in a binary file with the extension `dhc`. This information includes which IP addresses are allocated and the remaining lease time for the addresses. If the router is restarted, the DHC file ensures that the server retains this information for the clients it is currently serving.

Configuration Example

The following example shows how to configure a router to act as a DHCP server in a small site. The site has a limited range of IP addresses and users use IP for short periods of time. The dynamic DHCP mechanism is the most appropriate for this situation. The router on the LAN is configured to provide DHCP services to the PCs on the local LAN.

1. Enable the DHCP server.

To enable DHCP, use the command:

```
enable dhcp
```

2. Create a policy.

Use the following commands to create a policy that sets the base configuration information required by the client hosts:

```
create dhcp policy=base lease=7200
add dhcp policy=base subnet=255.255.255.0
add dhcp policy=base router=192.168.1.1
add dhcp policy=base dnsserver=192.168.1.254,
192.168.1.253
```

3. Create a range.

To create a range that defines a list of IP addresses where the policy applies, use the command:

```
create dhcp range=office policy=base ip=192.168.1.16
number=32
```

4. Test the configuration.

To check that DHCP is functioning correctly, use the commands:

```
show dhcp
show dhcp policy
show dhcp range
show dhcp client
```

5. Configure a printer.

To configure a printer with the MAC address of 00-00-0c-00-28-73 that talks BOOTP, use the commands:

```
enable dhcp bootp
create dhcp policy=prnt lease=infinity inherit=base
add dhcp range=office policy=prnt IP=192.168.1.31
address=00-00-0c-00-28-73
```

Troubleshooting

Problem The router is enabled as a DHCP server but has not assigned an IP address to a host.

- Solutions**
- Reboot the host machine, to force it to re-request IP settings.
 - Check the host's TCP/IP settings to make sure that the host is set to obtain its IP address dynamically:

In Windows 95/98, click Settings > Control Panel > Network. Select TCP/IP and click Properties. Click **Obtain an IP address automatically**.

In Windows 2000, click Settings > Control Panel > Network and Dial-up Connections > Local Area Connection > Properties. Select Internet connection (TCP/IP) and click Properties. Click **Obtain an IP address automatically**.
 - Check that the DHCP server has a large enough range of addresses. To use the GUI to do this, select Configuration > DHCP Server.
 - Check that the router's link to the LAN is functioning by checking the interface status, and verifying that the link LED is lit.

Command Reference

This section describes the commands available to configure and manage the Dynamic Host Configuration Protocol (DHCP) on the router.

DHCP requires the IP module to be enabled and configured correctly. See [Chapter 22, Internet Protocol \(IP\)](#) for detailed descriptions of commands that enable and configure IP.

The shortest valid command is denoted by capital letters in the Syntax section. See “Conventions” on page lxv of [About this Software Reference](#) in the front of this manual for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

add bootp relay

Syntax ADD BOOTp RELAY=*ipadd* [INTerface=*interface*]

Description This command adds a BOOTP relay destination. Note that if an interface has no relay destinations defined, the router relays packets received on that interface to any relay destination that does not apply to a specified interface.

Parameter	Description
RELAY	<p>The IP address of a BOOTP or DHCP server in dotted decimal notation. The <i>ipadd</i> is an IP address in dotted decimal notation. Up to 50 relay destinations can be defined. You can add an IP address multiple times, specifying different interface values.</p> <p>Default: no default</p>
INTerface	<p>The interface that this relay destination applies to. BOOTP packets received on this interface are relayed to the specified relay destination only. You can define the same interface for multiple relay destinations; the router relays any BOOTP packets received to each relay destination. The <i>interface</i> is an interface name formed by concatenating a Layer 2 interface type and an interface instance. Valid interfaces are:</p> <ul style="list-style-type: none"> ■ eth (such as eth0) ■ ATM (such as atm0.1) ■ PPP (such as ppp0) ■ FR (such as fr0) ■ X.25 DTE (such as x25t0) ■ VLAN (such as vlan1) <p>Note that you cannot specify the logical interface attached to an interface.</p> <p>Default: no default (if the router receives BOOTP messages on interfaces that have no specific relay destinations, it relays them to this destination)</p>

Examples To relay packets for the DHCP server with IP address 192.168.13.11, use:

```
add boot rela=192.168.13.11
```

Related Commands

- [delete bootp relay](#)
- [disable bootp relay](#)
- [enable bootp relay](#)
- [purge bootp relay](#)
- [set bootp relay maxhops](#)
- [show bootp relay](#)

add dhcp option

Syntax ADD DHCP OPTion=*number* POLIcy=*name*
 TYpe={IP|SWItch|VALue|STRing|HexString|NONE}
 VALue=*value* [NAME=*option-name*]

Description This command allows you to create a user-defined option for the specified policy. User-defined options are outside the standard range of pre-defined options that you can define using the [add dhcp policy command on page 23-13](#).

It is possible to add a user-defined option with the same number as an existing pre-defined option. If this situation occurs, the user-defined option takes precedence - that is, it overrides but does not eliminate the pre-defined option.

Parameter	Description														
OPTion	A number for the option. <i>number</i> is a decimal number between 1 and 254.														
POLIcy	The name of the policy to add the option to. <i>name</i> is a character string 1 to 15 characters long. Any printable character is allowed. When you enter a <i>name</i> that contains spaces, you must surround it with double quotation marks.														
NAME	Use this optional parameter to define a name for the option. <i>option-name</i> is a character string 1 to 15 characters long. Any printable character is allowed. When you enter an <i>option-name</i> that contains spaces, you must surround it with double quotation marks.														
TYpe	Use this optional parameter to specify a format in which to define the value parameter. If you specify a type , the value parameter is mandatory. Choose from the following type options:														
	<table border="1"> <thead> <tr> <th>TYpe</th> <th>Value format</th> </tr> </thead> <tbody> <tr> <td>IP</td> <td>One or more IPV4 addresses in dotted decimal format, separated by commas.</td> </tr> <tr> <td>SWItch</td> <td>Any of: on, off, yes, no, true, false, enabled, disabled.</td> </tr> <tr> <td>VALue</td> <td>A decimal number between 0 and 4294967295.</td> </tr> <tr> <td>STRing</td> <td>A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.</td> </tr> <tr> <td>HexString</td> <td>A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.</td> </tr> <tr> <td>NONE</td> <td>No value is required.</td> </tr> </tbody> </table>	TYpe	Value format	IP	One or more IPV4 addresses in dotted decimal format, separated by commas.	SWItch	Any of: on, off, yes, no, true, false, enabled, disabled .	VALue	A decimal number between 0 and 4294967295.	STRing	A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.	HexString	A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.	NONE	No value is required.
TYpe	Value format														
IP	One or more IPV4 addresses in dotted decimal format, separated by commas.														
SWItch	Any of: on, off, yes, no, true, false, enabled, disabled .														
VALue	A decimal number between 0 and 4294967295.														
STRing	A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.														
HexString	A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.														
NONE	No value is required.														
	Default: none														
VALue	<i>value</i> is a user-defined value, which you must enter in the format specified with the type parameter - see above for details.														

Examples To add option 151 to the 'base' policy with the **name** 'svpsrvr', and a **type** of **ip**, use the command:

```
add dhcp opt=151 poli=base nam=svpsrvr ty=ip
val=192.168.3.3
```

To add option 114 to the 'base' policy with no **name**, and a **type** of **string**, use the command:

```
add dhcp opt=114 poli=base ty=str val=http://allied-
telesis.com
```

Related Commands

- [add dhcp policy](#)
- [delete dhcp option](#)
- [set dhcp option](#)
- [show dhcp policy](#)

add dhcp policy

Syntax ADD DHCP POLIcy=*name* [ARPTIMEout=*seconds*]
 [BOOTFilesize=*bootfilesize*] [BROADCastaddress=*ipadd*]
 [COOKIeserver=*ipadd, ipadd...*]
 [DNSServer=*ipadd, ipadd...*] [DomaInname=*string*]
 [ETHERENcap={ON|OFF}] [EXTENSIOncap=*string*]
 [File=*string*] [HOSTName=*string*]
 [IMPRESSserver=*ipadd, ipadd...*] [INTMTu=68..65535]
 [IPForwarding={ENABLEd|DISabled}] [IPMTU=576..65535]
 [IPPLAteau=*mtu, mtu...*] [IPTIMEout=*seconds*] [IPTTL=*t1*]
 [LOGServer=*ipadd, ipadd...*] [LPRServer=*ipadd, ipadd...*]
 [MASKDiscovery={ON|OFF}] [MASKSupplier={ON|OFF}]
 [MERITdumpfile=*longstring*] [NAMEserver=*ipadd, ipadd...*]
 [NBDDservers=*ipadd, ipadd...*]
 [NBNameservers=*ipadd, ipadd...*] [NBNODEtype={B-node|
 P-node|M-node|H-node}] [NBSCOpe=*string*]
 [NISDomain=*string*] [NIServers=*ipadd, ipadd...*]
 [NTPServers=*ipadd, ipadd...*]
 [POLICYFiltering=*ipadd, ipadd...*]
 [RESOURceserver=*ipadd, ipadd...*] [ROOTPath=*longstring*]
 [ROUTer=*ipadd, ipadd...*] [ROUTERDiscovery={ON|OFF}]
 [ROUTERSolicit=*ipadd*] [SERVER=*ipadd*]
 [SERVERName=*server-name*] [SOURCErouting={ENABLEd|
 DISabled}] [STATICroute=*ipadd, ipadd...*] [SUBLOCAL={ON|
 OFF}] [SUBNetmask=*ipadd*] [SWAPServer=*ipadd*]
 [T1TIme=*seconds*] [T2TIme=*seconds*] [TCPGarbage={ON|OFF}]
 [TCPKeepalive=*seconds*] [TCPTtl=*t1*]
 [TIMEOffset=*utc-offset*] [TIMEServer=*ipadd, ipadd...*]
 [TRAILerencap={ON|OFF}]
 [XDISplayservers=*ipadd, ipadd...*]
 [XFONTservers=*ipadd, ipadd...*]

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *seconds* is a time, time offset or timeout value in seconds.
- *bootfilesize* is the length in 512-octet blocks of the default boot image for the client.
- *ipadd* is an IP address in dotted decimal notation.
- *string* is a character string 1 to 99 characters long. It may contain any printable character.
- *longstring* is a character string 1 to 254 characters long. It may contain any printable character.
- *t1* is a number from 1 and 255.
- *server-name* is a character string 1 to 63 characters long. It may contain any printable character.
- *utc-offset* is a time offset in seconds from Coordinated Universal Time (UTC).

Description This command adds a standard pre-defined option to an existing DHCP policy. Each parameter has an associated IANA allocated option number which it defines, this is shown in brackets after the parameter description.

For more information about IANA allocated options, see <http://www.iana.org/assignments/bootp-dhcp-parameters>.

You can override any of these options by creating a user-defined option with the same number using the **add dhcp option** command on page 23-11.

The **policy** parameter specifies the name of the policy where the option is to be added.

The **arptimeout** parameter specifies the timeout in seconds for ARP cache entries (Option 35).

The **bootfilesize** parameter specifies the length in 512-octet blocks of the default boot image for the client (Option 13).

The **broadcastaddress** parameter specifies the broadcast address in use on the client's subnet (Option 28).

The **cookieserver** parameter specifies a list of RFC 865 cookie servers available to the client. Cookie servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 8).

The **dnserver** parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Domain Name System name servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 6).

The **domainname** parameter specifies the domain name that the client should use when resolving host names via the Domain Name System (Option 15).

The **etherencap** parameter specifies whether the client should use Ethernet Version 2 (RFC 894) or IEEE Standard 802.3 (RFC 1042) encapsulation for Ethernet interfaces (eth and vlan). A value of **off** indicates that the client should use RFC 894 encapsulation. A value of **on** means that the client should use RFC 1042 encapsulation (Option 36).

The **extensionpath** parameter specifies a string to specify a file, retrievable via TFTP, which contains information that can be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response (Option 18).

The **file** parameter specifies the boot file name for the client (Option 67).

The **hostname** parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions (Option 12).

The **impressserver** parameter specifies a list of Imagen Impress servers available to the client. Imagen Impress servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 10).

The **intmtu** parameter specifies the interface's Maximum Transmission Unit (MTU), in bytes. This is the maximum packet size that the client can transmit over this interface at the physical layer (for example, Ethernet). Higher-layer packets (for example, IP datagrams) that are bigger than this is fragmented by the client and reassembled at the receiving end (Option 26).

The **ipforwarding** parameter specifies whether the client should configure its IP layer for packet forwarding. A value of **disable** disables IP forwarding, and a value of **enable** enables IP forwarding (Option 19).

The **ipmtu** parameter specifies the size, in bytes, of the largest IP packet the client should be prepared to reassemble from packets that were fragmented at the physical layer. The client discards IP packets that are larger than this value (Option 22).

The **ipplateau** parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68 (Option 25).

The **iptimeout** parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 (Option 24).

The **ipttl** parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255 (Option 23).

The **logserver** parameter specifies a list of MIT-LCS UDP log servers available to the client. Log servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 7).

The **lprserver** parameter specifies a list of RFC 1179 line printer servers available to the client. Line printer servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 9).

The **maskdiscovery** parameter specifies whether the client should perform subnet mask discovery using ICMP. A value of OFF indicates that the client should not perform mask discovery. A value of ON means that the client should perform mask discovery (Option 29).

The **masksupplier** parameter specifies whether the client should respond to subnet mask requests using ICMP. A value of OFF indicates that the client should not respond. A value of ON means that the client should respond (Option 30).

The **meritdumpfile** parameter specifies the path name of a file where the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 14).

The **nameserver** parameter specifies a list of IEN116 name servers available to the client. IEN116 servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 5).

The **nbdbservers** parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference. NetBIOS datagram distribution servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 45).

The **nbnameservers** parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference. NetBIOS name servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 44).

The **nbnodetype** parameter specifies the NetBIOS node type that allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002 (Option 46).

The **nbscope** parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002 (Option 47).

The **nisdomain** parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set (Option 40).

The **niservers** parameter specifies a list of IP addresses indicating NIS servers available to the client. NIS servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 41).

The **ntpervers** parameter specifies a list of IP addresses indicating NTP servers available to the client. NTP servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 42).

The **policyfiltering** parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks that specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client. Policy filters are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 21).

The **resourceserver** parameter specifies a list of RFC 887 Resource Location servers available to the client. Resource Location servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 11).

The **rootpath** parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 17).

The **router** parameter specifies a list of IP addresses for routers on the client's subnet. Routers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Routers should be listed in order of preference (Option 3).

The **routerdiscovery** parameter specifies whether the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of **off** indicates that the client should not perform router discovery. A value of **on** means that the client should perform router discovery (Option 31).

The **routersolicit** parameter specifies the address where the client should transmit router solicitation requests (Option 32).

The **server** parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set (Option 65).

The **servername** parameter specifies the name of the server host. This is passed to the client (Option 66).

The **sourcerouting** parameter specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of **disable** disallows forwarding of such datagrams, and a value of **enable** allows forwarding (Option 20).

The **staticroute** parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. A maximum of up to 32 IP addresses can be specified in a comma separated list. The default route (0.0.0.0) is an illegal destination for a static route (Option 33).

The **sublocal** parameter specifies whether the client may assume that all subnets of the IP network where the client is connected use the same MTU as the subnet of that network where the client is directly connected. A value of **on** indicates that all subnets share the same MTU. A value of **off** means that the client should assume that some subnets of the directly connected network may have smaller MTUs (Option 27).

The **subnetmask** parameter specifies the client's subnet mask as defined in RFC 950. If you intend to use subnet or supernet addressing when you create DHCP ranges, you should include this option in the policy before you create the range (Option 1).

The **swapservers** parameter specifies the IP address of the client's swap server (Option 16).

The **t1time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **renewing** state (Option 58).

The **t2time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **rebinding** state (Option 59).

The **tcpgarbage** parameter specifies whether the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of **off** indicates that a garbage octet should not be sent. A value of **on** indicates that a garbage octet should be sent (Option 39).

The **tcpkeepalive** parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application (Option 38).

The **tcpttl** parameter specifies the default time-to-live value that the client should use when sending TCP segments (Option 37).

The **timeoffset** parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC) (Option 2).

The **timeserver** parameter specifies a list of RFC 868 time servers available to the client. Time servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 4).

The **trailerencap** parameter specifies whether the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of **off** indicates that the client should not attempt to use trailers. A value of **on** means that the client should attempt to use trailers (Option 34).

The **xdisplayserver**s parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. A maximum of up to 32 IP addresses can be specified in a comma separated list. Addresses should be listed in order of preference (Option 49).

The **xfontserver**s parameter specifies a list of X Window System Font servers available to the client. X Window System Font servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 48).

Examples To create a policy called "base" with subnet mask, router and DNS server options, use the command:

```
add dhcp poli=base subn=255.255.255.0 rou=202.36.163.21
    dns=192.168.100.50,192.168.100.33
```

Related Commands

- [add dhcp option](#)
- [create dhcp policy](#)
- [delete dhcp policy](#)
- [destroy dhcp policy](#)
- [set dhcp policy](#)
- [show dhcp policy](#)

add dhcp range

Syntax `ADD DHCP RANge=name Address=macadd IP=ipadd [POLIcy=name]`

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.
- *macadd* is a hardware address of the form *xx-xx-xx-xx-xx-xx*, where *xx* is a 2-digit hexadecimal number with leading zeros if necessary.

Description This command adds a static entry to an existing DHCP range. The **range** parameter specifies the name of an existing DHCP range.

The **ip** parameter specifies the IP address of the host to add to the range. The **address** parameter defines the MAC address for the static host entry. The **policy** parameter specifies the name of a policy to give the host entry.

Examples To add a static entry to the range “remote” for the device with MAC address 00-00-0c-00-28-73, use the command:

```
add dhcp ran=remote ip=192.168.1.31 a=00-00-0c-00-28-73
```

Related Commands

- [create dhcp range](#)
- [delete dhcp range](#)
- [destroy dhcp range](#)
- [set dhcp range](#)
- [show dhcp range](#)

create dhcp policy

Syntax CREate DHCP POLIcy=*name* LEASEtime={*lease-time*|INFinity}
[INHerit=*name*]

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *lease-time* is a time in seconds from 1 to 4294967295.

Description This command creates a DHCP policy. Policies define the configuration information that are given to the requesting IP host. The **policy** parameter specifies the name of the policy to create. This name is used in other commands to identify the policy.

The **leasetime** parameter specifies the time period for which the IP address is leased to the requesting IP client. For BOOTP requests, **leasetime** must be set to **infinity**. If dynamic IP address allocation is not required, then set **leasetime** to **infinity**.

Routers that comply with RFC 1541 but not with RFC 2131 may not be able to accept a lease time of less than 3600 seconds (1 hour).

The **inherit** parameter specifies the name of an existing policy whose settings are inherited by the new policy. This parameter allows the building of hierarchical policies and reduces the number of commands to create similar policies.

Examples To create a DHCP policy called “base” with a default lease time of two hours, use the command:

```
cre dhcp poli=base lease=7200
```

Related Commands

- [add dhcp option](#)
- [add dhcp policy](#)
- [delete dhcp policy](#)
- [destroy dhcp policy](#)
- [set dhcp policy](#)
- [show dhcp policy](#)

create dhcp range

Syntax `CREate DHCP RANge=name IP=ipadd NUMber=number POLIcy=name
[GATEway=ipadd] [PRObe={ARP|ICMP}]`

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.
- *number* is a number from 1 to 255.

Description This command creates a DHCP range. The server tries to fulfil BOOTP or DHCP requests from hosts with IP addresses in the defined ranges. The **range** parameter specifies the name of the range to create.

The **policy** parameter specifies the name of a default policy to give the range. Individual host entries in the range can later be set to other defined policies.

The **ip** parameter defines the IP address of the start of the range.

If the range you specify includes network or broadcast addresses, these are not added to the pool of available addresses.

The **number** parameter defines how many host entries from the start IP address are to be included in the range.

The **gateway** parameter specifies the IP address of a remote BOOTP relay agent. This parameter is needed if the range of IP addresses specified are not on a local interface.

The **probe** parameter specifies how the DHCP server checks whether an IP address is being used by other hosts. If **arp** is specified, the server sends ARP requests to determine if an address is in use. If **icmp** is specified, the server sends ICMP Echo Requests (pings). **arp** cannot be specified if the **gateway** parameter is specified or if the network uses Proxy ARP. The default is **icmp**.

Examples To create a range called “office”, which uses the policy “base”, with 32 IP addresses starting at 192.168.1.16, use the command:

```
cre dhcp ran=office poli=base ip=192.168.1.16 num=32
```

Related Commands

- [add dhcp range](#)
- [delete dhcp range](#)
- [destroy dhcp range](#)
- [set dhcp range](#)
- [show dhcp range](#)

delete bootp relay

Syntax DELEte BOOTp RELAY=*ipadd* [INTErface=*interface*]

Description This command deletes a BOOTP relay destination.

Parameter	Description
RELAY	The IP address of a BOOTP or DHCP server in dotted decimal notation. The <i>ipadd</i> is an IP address in dotted decimal notation. Default: no default
INTErface	The interface that this relay destination applies to. You must specify interface when deleting any relay destination that was added with a defined interface. The <i>interface</i> is an interface name formed by concatenating a Layer 2 interface type and an interface instance. Valid interfaces are: <ul style="list-style-type: none"> ■ eth (such as eth0) ■ ATM (such as atm0.1) ■ PPP (such as ppp0) ■ FR (such as fr0) ■ X.25 DTE (such as x25t0) ■ VLAN (such as vlan1) Note that you cannot specify the logical interface attached to an interface. Default: no default

Examples To delete the BOOTP server with IP address 192.168.13.11, use:

```
del boot rela=192.168.13.11
```

Related Commands

- [add bootp relay](#)
- [disable bootp relay](#)
- [enable bootp relay](#)
- [purge bootp relay](#)
- [set bootp relay maxhops](#)
- [show bootp relay](#)

delete dhcp option

Syntax `DELEte DHCP OPTion=number POLIcy=name`

Description This command deletes a user-defined option from the specified policy. User-defined options are created with the [add dhcp option command on page 23-11](#).

It is possible for the same option number to be specified for different options, one using **add dhcp option**, and one using **add dhcp policy**. This command deletes only the option created with **add dhcp option**.

To completely delete the option number from the system, you must also delete the option with the same number that was created with **add dhcp policy**. You can do this using the **delete dhcp policy** command.

Once this option is deleted, any existing pre-defined option with the same option number becomes the active option.

Parameter	Description
OPTion	The number of the option to delete. This option must have been defined using add dhcp option . <i>number</i> is a decimal number between 1 and 254.
POLIcy	The name of the policy that the option is attached to. <i>name</i> is a character string 1 to 15 characters long. It may contain any printable character.

Example To delete option 151 from the 'base' policy, use the command:

```
del dhcp opt=151 poli=base
```

Related Commands

- [add dhcp option](#)
- [add dhcp policy](#)
- [delete dhcp policy](#)
- [set dhcp option](#)
- [show dhcp policy](#)

delete dhcp policy

Syntax DELEte DHCP POLIcy=*name* [ARPTIMEout] [BOOTFilesize] [BROADcastaddress] [COOKIeserver] [DNSServer] [Domainname] [ETHEREncap] [EXTENSIONpath] [File] [HOSTname] [IMPRESSserver] [INTMTu] [IPForwarding] [IPMTU] [IPPLAteau] [IPTIMEout] [IPTTL] [LOGServer] [LPRServer] [MASKDiscovery] [MASKSupplier] [MERITdumpfile] [NAMEserver] [NBDDservers] [NBNameservers] [NBNOdetype] [NBSCOpe] [NISDomain] [NIServers] [NTPServers] [POLICYFiltering] [RESOURceserver] [ROOTPath] [ROUTer] [ROUTERDiscovery] [ROUTERSolicit] [SERVER] [SERVERName] [SOURCErouting] [STATicroute] [SUBLOCAL] [SUBNetmask] [SWAPServer] [T1Time] [T2Time] [TCPGarbage] [TCPKeepalive] [TCPttl] [TIMEOffset] [TIMEServer] [TRAILerencap] [XDISplayservers] [XFONTservers]

where *name* is a character string 1 to 15 characters long. It may contain any printable character.

Description This command deletes an existing pre-defined option, added using **add dhcp policy**, from an existing DHCP policy. Each parameter has an associated IANA allocated option number which it defines, this is shown in brackets after the parameter description.

The **policy** parameter specifies the name of the policy from which the option is to be removed.

The **arptimeout** parameter specifies the timeout in seconds for ARP cache entries (Option 35).

The **bootfilesize** parameter specifies the length in 512-octet blocks of the default boot image for the client (Option 13).

The **broadcastaddress** parameter specifies the broadcast address in use on the client's subnet (Option 28).

The **cookieserver** parameter specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference. (Option 8).

The **dnsserver** parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers should be listed in order of preference (Option 6).

The **domainname** parameter specifies the domain name that the client should use when resolving host names via the Domain Name System (Option 15).

The **etherencap** parameter specifies whether the client should use Ethernet Version 2 (RFC 894) or IEEE Standard 802.3 (RFC 1042) encapsulation for Ethernet interfaces (eth and vlan). A value of **off** indicates that the client should use RFC 894 encapsulation. A value of **on** means that the client should use RFC 1042 encapsulation (Option 36).

The **extensionpath** parameter specifies a string to specify a file, retrievable via TFTP, which contains information that can be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response (Option 18).

The **file** parameter specifies the boot file name for the client (Option 67).

The **hostname** parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions (Option 12).

The **impressserver** parameter specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference (Option 10).

The **intmtu** parameter specifies the interface's Maximum Transmission Unit (MTU), in bytes. This is the maximum packet size that the client can transmit over this interface at the physical layer (for example, Ethernet). (Option 26).

The **ipforwarding** parameter specifies whether the client should configure its IP layer for packet forwarding. A value of **disable** disables IP forwarding, and a value of **enable** enables IP forwarding (Option 19).

The **ipmtu** parameter specifies the size, in bytes, of the largest IP packet the client should be prepared to reassemble. (Option 22).

The **ipplateau** parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68 (Option 25).

The **iptimeout** parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 (Option 24).

The **ipttl** parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255 (Option 23).

The **logserver** parameter specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference (Option 7).

The **lprserver** parameter specifies a list of RFC 1179 line printer servers available to the client. Servers should be listed in order of preference (Option 9).

The **maskdiscovery** parameter specifies whether the client should perform subnet mask discovery using ICMP. A value of **off** indicates that the client should not perform mask discovery. A value of **on** means that the client should perform mask discovery (Option 29).

The **masksupplier** parameter specifies whether the client should respond to subnet mask requests using ICMP. A value of **OFF** indicates that the client should not respond. A value of **ON** means that the client should respond (Option 30).

The **meritdumpfile** parameter specifies the path name of a file where the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 14).

The **nameserver** parameter specifies a list of IEN116 name servers available to the client. Servers should be listed in order of preference (Option 5).

The **nbddservers** parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference (Option 45).

The **nbnameservers** parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference (Option 44).

The **nbnodetype** parameter specifies the NetBIOS node type that allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002 (Option 46).

The **nbscope** parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002 (Option 47).

The **nisdomain** parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set (Option 40).

The **niservers** parameter specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference (Option 41).

The **ntpervers** parameter specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference (Option 42).

The **policyfiltering** parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks that specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client. (Option 21).

The **resourceserver** parameter specifies a list of RFC 887 Resource Location servers available to the client. Servers should be listed in order of preference (Option 11).

The **rootpath** parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 17).

The **router** parameter specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference (Option 3).

The **routerdiscovery** parameter specifies whether the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of **off** indicates that the client should not perform router discovery. A value of **on** means that the client should perform router discovery (Option 31).

The **routersolicit** parameter specifies the address where the client should transmit router solicitation requests (Option 32).

The **server** parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set (Option 65).

The **servername** parameter specifies the name of the server host. This is passed to the client (Option 66).

The **sourcerouting** parameter specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of **disable** disallows forwarding of such datagrams, and a value of **enable** allows forwarding (Option 20).

The **staticroute** parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route (Option 33).

The **sublocal** parameter specifies whether the client may assume that all subnets of the IP network where the client is connected use the same MTU as the subnet of that network where the client is directly connected. A value of **on** indicates that all subnets share the same MTU. A value of **off** means that the client should assume that some subnets of the directly connected network may have smaller MTUs (Option 27).

The **subnetmask** parameter specifies the client's subnet mask as defined in RFC 950. If you intend to use subnet or supernet addressing when you create DHCP ranges, you should include this option in the policy before you create the range (Option 1).

The **swapservers** parameter specifies the IP address of the client's swap server (Option 16).

The **t1time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **renewing** state (Option 58).

The **t2time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **rebinding** state (Option 59).

The **tcpgarbage** parameter specifies whether the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of **off** indicates that a garbage octet should not be sent. A value of **on** indicates that a garbage octet should be sent (Option 39).

The **tcpkeepalive** parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application (Option 38).

The **tcpctl** parameter specifies the default time-to-live value that the client should use when sending TCP segments (Option 37).

The **timeoffset** parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC) (Option 2).

The **timeservers** parameter specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference (Option 4).

The **trailerencap** parameter specifies whether the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of **off** indicates that the client should not attempt to use trailers. A value of **on** means that the client should attempt to use trailers (Option 34).

The **xdisplayserver**s parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. A maximum of up to 32 IP addresses can be specified in a comma separated list. (Option 49).

The **xfontserver**s parameter specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference (Option 48).

Examples To remove the LPRSERVER option from the policy “base”, use the command:

```
del dhcp poli=base lprs
```

Related Commands

- [add dhcp policy](#)
- [create dhcp policy](#)
- [delete dhcp option](#)
- [destroy dhcp policy](#)
- [set dhcp policy](#)
- [show dhcp policy](#)

delete dhcp range

Syntax DELEte DHCP RANge=*name* IP=*ipadd*

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes an entry for a static or active dynamic host from a DHCP range. The entry goes through the reclaim process and returns to dynamic use if the client is still using the address.

The **range** parameter specifies the name of the range. The **ip** parameter specifies the host entry to remove.

Examples To delete the static entry 192.168.1.31 from the range “remote”, use the command:

```
del dhcp ran=remote ip=192.168.1.31
```

Related Commands

- [add dhcp range](#)
- [create dhcp range](#)
- [destroy dhcp range](#)
- [show dhcp range](#)

destroy dhcp policy

Syntax DESTroy DHCP POLIcy=*name*

where *name* is a character string 1 to 15 characters long. It may contain any printable character.

Description This command destroys an existing policy. The **policy** parameter specifies the name of the policy to destroy. If the policy is currently being used by any host entry, then an error message is displayed and the command fails.

Examples To destroy policy “admin”, use the command:

```
dest dhcp poli=admin
```

Related Commands [add dhcp policy](#)
[create dhcp policy](#)
[delete dhcp policy](#)
[set dhcp policy](#)
[show dhcp policy](#)

destroy dhcp range

Syntax DESTroy DHCP RANge=*name*

where *name* is a character string 1 to 15 characters long. It may contain any printable character.

Description This command destroys an existing DHCP range. The **range** parameter specifies the name of the range to destroy.

Examples To destroy the range “remote”, use the command:

```
dest dhcp ran=remote
```

Related Commands [add dhcp range](#)
[create dhcp range](#)
[delete dhcp range](#)
[show dhcp range](#)

disable bootp relay

Syntax DISable BOOTp RELAY

Description This command disables the BOOTP Relay Agent. The BOOTP Relay Agent relays BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination, and relays BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router. BOOTREPLY messages addressed to clients on networks not directly connected to the router are ignored by the relay agent and treated as ordinary IP packets for forwarding. The BOOTP Relay Agent is disabled by default.

Examples To disable the BOOTP relay agent, use the command:

```
dis boot rela
```

Related Commands [add bootp relay](#)
[delete bootp relay](#)
[enable bootp relay](#)
[purge bootp relay](#)
[set bootp relay maxhops](#)
[show bootp relay](#)

disable bootp relay option82

Syntax DISable BOOTp RELAY OPTion82 [DEBug]

Description This command disables the insertion of DHCP Option 82 into the DHCP options field when forwarding client-originated BOOTP packets to a DHCP or BOOTP server.

Use the **debug** parameter to disable Option 82 related debug.

Example To disable the insertion of Option 82, use the command:

```
dis boot rela opt
```

Related commands [enable bootp relay option82](#)
[set bootp relay option82](#)
[set bootp relay option82 port](#)
[show bootp relay port](#)
[disable bootp relay](#)

disable dhcp

Syntax DISable DHCP [BOOTp]

Description This command disables the DHCP module. The router ignores all BOOTP or DHCP requests that are received while the module is disabled.

The **bootp** parameter stops the router from processing BOOTP requests. This means BOOTP clients cannot request IP addresses. DHCP client requests are still processed by the router.

Examples To disable the DHCP server, use the command:

```
dis dhcp
```

To disable BOOTP service, use the command:

```
dis dhcp boot
```

Related Commands [enable dhcp](#)
[show dhcp](#)

disable dhcp debug

Syntax `DISable DHCP DEBug={ALL|PKT|RECLaim|STAtE}`

Description This command disables the DHCP module. The router ignores all BOOTP or DHCP requests that are received while the module is disabled.

As well as disabling DHCP, you can also use this command to disable other DHCP functions, using the optional parameters listed below:

Parameter	Description
DEBug	The specific DHCP debugging mode that is disabled. Default: no default
ALL	All debugging modes are disabled.
PKT	Incoming and outgoing packet debugging is disabled.
RECLaim	DHCP debugging relating to DHCP address reclaiming is disabled.
STAtE	State debugging is disabled. DHCP processing status information is no longer displayed.

Examples To disable packet debugging, use the command:

```
dis dhcp deb=pkt
```

Related Commands [enable dhcp debug](#)
[show dhcp](#)

enable bootp relay

Syntax `ENAbLe BOOTp RELAY`

Description This command enables the BOOTP Relay Agent. The BOOTP Relay Agent relays BOOTREQUEST messages originating from any of the router's interfaces to a user-defined destination, and relays BOOTREPLY messages addressed to BOOTP clients on networks directly connected to the router. BOOTREPLY messages addressed to clients on networks not directly connected to the router are ignored by the relay agent and treated as ordinary IP packets for forwarding. The BOOTP Relay Agent is disabled by default.

Related Commands [add bootp relay](#)
[delete bootp relay](#)
[disable bootp relay](#)
[purge bootp relay](#)
[set bootp relay maxhops](#)
[show bootp relay](#)

enable bootp relay option82

Syntax ENAbLe BOOTp RELAY OPTion82 [DEBug]

Description This command enables the DHCP relay agent to insert DHCP Option 82 into the DHCP options field when forwarding client-originated BOOTP/DHCP packets to a DHCP server.

Use the **debug** parameter to enable Option 82 related debug.

Example To enable the insertion of Option 82, use the command:

```
ena boot rela opt
```

Related Commands [disable bootp relay option82](#)
[set bootp relay option82](#)
[set bootp relay option82 port](#)
[enable bootp relay](#)
[show bootp relay port](#)

enable dhcp

Syntax ENAbLe DHCP [BOOTp]

Description This command enables the DHCP module.

The **bootp** parameter enables the switch to process all BOOTP requests, as well as DHCP requests. This allows BOOTP clients to requests IP addresses.

Examples To enable the DHCP server, use the command:

```
ena dhcp
```

To enable BOOTP service, use the command:

```
ena dhcp boot
```

Related Commands [disable dhcp](#)
[show dhcp](#)

enable dhcp debug

Syntax ENABle DHCP DEBUg={ALL|PKT|RECLaim|STAtE}

Description This command enables debugging for DHCP.

Parameter	Description
DEBUg	The specific DHCP debugging mode that is enabled. The router sends debugging information to the port or Telnet session where you entered the command. Default: no default
ALL	All debugging modes are enabled.
PKT	Incoming and outgoing packet debugging is enabled.
RECLaim	DHCP debugging relating to DHCP address reclaiming is enabled. Note that this can be repetitive if you have a large configuration, and especially if you have just enabled DHCP. We therefore recommend that you only enter this option if you intend to monitor or debug reclaiming activity.
STAtE	State debugging is enabled. DHCP processing status information is now displayed.

Examples To enable all debugging modes, use one of the commands:

```
ena dhcp deb=all
```

Related Commands [disable dhcp debug](#)
[show dhcp](#)

purge bootp relay

Syntax PURge BOOTp RELAY

Description This command purges the BOOTP relay configuration. The BOOTP relay agent is disabled and all configuration data is purged.

Related Commands [add bootp relay](#)
[delete bootp relay](#)
[disable bootp relay](#)
[enable bootp relay](#)
[set bootp relay maxhops](#)
[show bootp relay](#)

set bootp relay maxhops

Syntax SET BOOTp RELAY MAXHops=1..16

Description This command sets the hop count threshold for discarding BOOTP messages. When the hops field in a BOOTP message exceeds the threshold, the BOOTP message is discarded. The hop count in a BOOTP message is incremented each time a router forwards the message. The default is 4.

The syntax **set bootp maxhops** is a synonym of **set bootp relay maxhops**.

Related Commands

- [add bootp relay](#)
- [delete bootp relay](#)
- [disable bootp relay](#)
- [enable bootp relay](#)
- [purge bootp relay](#)
- [show bootp relay](#)

set bootp relay option82

Syntax SET BOOTp RELAY OPTion82 [CHEck={YES|NO|ON|OFF|True|False}] [POLIcy={DROP|KEEP|REPLACE}]

Description This command defines the checking and re-forwarding settings used by DHCP Option 82. When Option 82 is enabled, the DHCP relay agent inserts Option 82 information into the DHCP options field when forwarding client-originated BOOTP/DHCP packets to a DHCP server. Option 82 must be enabled with the **enable bootprelay option 82** command for the settings you specify to take effect.

Use the **check** parameter to specify whether the Option 82 information that is returned from the DHCP server is to be checked or not. When checking is enabled, server DHCP packets that contain valid Option 82 information are forwarded to the client, and packets that do not contain valid Option 82 information are dropped. If **yes** is specified, checking is enabled. The values **yes**, **on**, and **true** are equivalent. If **no** is specified, Option 82 information returned from the DHCP server is not checked. The values **no**, **off**, and **false** are equivalent. The default is **yes**.

Use the **policy** parameter to specify the re-forwarding policy of client DHCP packets that contain Option 82 information. If **drop** is specified, client DHCP packets that contain Option 82 information are dropped. If **keep** is specified, the packet keeps its existing Option 82 information. If **replace** is specified, the existing Option 82 information is replaced with that of the local device. The default is **replace**.

Example To set the re-forwarding policy to drop client DHCP packets with Option 82 information, use the command:

```
set boot rela opt poli=drop
```

Related commands

- [disable bootp relay option82](#)
- [enable bootp relay option82](#)
- [set bootp relay option82 port](#)
- [show bootp relay port](#)

set bootp relay option82 port

Syntax SET BOOTp RELAY OPTion82 POrt={*port-list*|ALL}
[SUBScriberid=*subscriber-id*] [TRusted={YES|NO|ON|OFF|
True|False}]

where:

- *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *subscriber-id* is a character string from 0 to 50 characters long. Valid characters are any alphanumeric characters. If string contains spaces, it must be in double quotes. Wildcards are not allowed.

Description This command defines the DHCP Relay Agent port settings for DHCP Option 82. When Option 82 is enabled, the Relay Agent inserts Option 82 information into the DHCP options field when forwarding client-originated BOOTP/DHCP packets to a DHCP server. Option 82 must be enabled with the **enable bootprelay option 82** command for the port settings you specify to take effect.

Use the **port** parameter to specify the port to use for this command. If **all** is specified, this command is applied to all ports on the device.

Use the **subscriberid** parameter to specify the subscriber-ID for the port defined with **port**. If specified, the subscriber-ID sub-option is included in the Option 82 field of client DHCP packets received on the specified port. The default is no subscriber-ID.

If you specify an empty string in the **subscriberid** parameter, then the subscriber-ID sub-option is not included in the Option 82 field of client DHCP packets forwarded from the specified port. Use this method to delete a subscriber-ID from a port.

Use the **trusted** parameter to specify how the router handles client DHCP packets that contain Option 82 information, but which have the giaddr field set to 0. If you specify **yes**, the defined port is considered to be a trusted source of Option 82 information, and packets with Option 82 information and a giaddr of 0 are forwarded according to normal BOOTP Relay operation. The values **yes**, **on**, and **true** are equivalent. If you specify **no**, packets are dropped that contain DHCP Option 82 information and with the giaddr field set to 0. The values **no**, **off**, and **false** are equivalent. The default is **no**.

Example To set all ports as trusted, use the command:

```
set boot rela opt po=all tr=yes
```

Related Commands [disable bootp relay option82](#)
[enable bootp relay option82](#)
[show bootp relay port](#)
[set bootp relay option82](#)

set dhcp

Syntax SET DHCP EXTendid={ON|OFF}

Description This command sets the DHCP extended identification option. An extended client ID is used when connecting multiple router interfaces to the same DHCP server.

This command affects new DHCP clients only. If extended DHCP identification is needed, then **dhcp extendid** must be on before DHCP clients are created. DHCP clients with this option set to **on** are incompatible with static DHCP server entries because they do not match the extended ID. To set DHCP clients see the [add ip interface command on page 22-80 of Chapter 22, Internet Protocol \(IP\)](#).

The **extendid** parameter specifies whether DHCP clients use an extended client ID when communicating with a DHCP server. If **off** is specified, the client ID value is the hardware address of the client interface. If **on** is specified, the client ID value is extended to include an internal interface identifier, uniquely distinguishing different interfaces on a device. The default is **off**.

Examples To enable the use of extended client ID values when connecting a multiple switch interfaces to the same DHCP server, use the command:

```
set dhcp ext=on
```

Related Commands [show dhcp](#)

set dhcp option

Syntax SET DHCP OPTion=*number* POLIcy=*name* [NAME=*option-name*]
 [TYpe={IP|SWItch|VALue|STRing|HexString|NONE}]
 [VALue=*value*]

Description This command allows you to modify an existing user-defined option on the specified policy. User-defined options are created using the [add dhcp option command on page 23-11](#).

You can modify the values set for the **name**, **type**, and **value** parameters. You cannot change the **policy** to which the option applies.

Parameter	Description														
OPTion	The number of the user-defined option to modify. <i>number</i> is a decimal number between 1 and 254														
POLlcy	The policy to which the option applies. <i>name</i> is a character string 1 to 15 characters long. Any printable character is allowed. When you enter a <i>name</i> that contains spaces, you must surround it with double quotation marks.														
NAME	Use this optional parameter to set a new name for the option. <i>option-name</i> is a character string 1 to 15 characters long. Any printable character is allowed. When you enter an <i>option-name</i> that contains spaces, you must surround it with double quotation marks.														
TYpe	Use this optional parameter to specify a format in which to define the value parameter. If you specify a type , the value parameter is mandatory. Choose from the following type options: <table border="1" data-bbox="715 1198 1426 1765"> <thead> <tr> <th>TYpe</th> <th>Value format</th> </tr> </thead> <tbody> <tr> <td>IP</td> <td>One or more IPV4 addresses in dotted decimal format, separated by commas.</td> </tr> <tr> <td>SWItch</td> <td>Any of: on, off, yes, no, true, false, enabled, disabled.</td> </tr> <tr> <td>VALue</td> <td>A decimal number between 0 and 4294967295.</td> </tr> <tr> <td>STRing</td> <td>A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.</td> </tr> <tr> <td>HexString</td> <td>A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.</td> </tr> <tr> <td>NONE</td> <td>No value is required.</td> </tr> </tbody> </table> Default: none	TYpe	Value format	IP	One or more IPV4 addresses in dotted decimal format, separated by commas.	SWItch	Any of: on, off, yes, no, true, false, enabled, disabled.	VALue	A decimal number between 0 and 4294967295.	STRing	A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.	HexString	A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.	NONE	No value is required.
TYpe	Value format														
IP	One or more IPV4 addresses in dotted decimal format, separated by commas.														
SWItch	Any of: on, off, yes, no, true, false, enabled, disabled.														
VALue	A decimal number between 0 and 4294967295.														
STRing	A character string from 1 to 255 characters long. Any printable character is allowed. When you enter a string that contains spaces, you must surround it with double quotation marks.														
HexString	A string of 1 to 255 sets of hexadecimal character pairs; a maximum of 510 characters. The 510 character maximum includes any blank spaces or quotes used.														
NONE	No value is required.														
VALue	<i>value</i> is a user-defined value, which you must enter in the format specified with the type parameter - see above for details. If you specify a value then the type parameter is mandatory.														

Examples To set a new name of 'server1' for option 151 on the 'base' policy, use the command:

```
set dhcp opt=151 poli=base nam=server1
```

To change the IP address for the user-defined option 151 on the 'base' policy to 192.168.3.2, use the command:

```
set dhcp opt=151 poli=base ty=ip value=192.168.3.2
```

Related Commands

- [add dhcp option](#)
- [delete dhcp option](#)
- [show dhcp policy](#)

set dhcp policy

Syntax SET DHCP POLICY=*name* [ARPTIMEout=*seconds*]
 [BOOTFilesize=*bootfilesize*] [BROADCASTaddress=*ipadd*]
 [COOKIEServer=*ipadd, ipadd...*]
 [DNSServer=*ipadd, ipadd...*] [DOMAINname=*string*]
 [ETHERENcap={ON|OFF}] [EXTENSIONpath=*string*]
 [File=*string*] [HOSTname=*string*]
 [IMPRESSServer=*ipadd, ipadd...*] [INTMTu=68..65535]
 [IPForwarding={ENABLED|DISABLED}] [IPMTU=576..65535]
 [IPPLAteau=*mtu, mtu...*] [IPTIMEout=*seconds*] [IPTTL=*t1*]
 [LEASETIME={*lease-time*|INFINITY}]
 [LOGServer=*ipadd, ipadd...*] [LPRServer=*ipadd, ipadd...*]
 [MASKDiscovery={ON|OFF}] [MASKSupplier={ON|OFF}]
 [MERITdumpfile=*longstring*] [NAMEserver=*ipadd, ipadd...*]
 [NBDDservers=*ipadd, ipadd...*]
 [NBNameservers=*ipadd, ipadd...*] [NBNOdetype={B-node|
 P-node|M-node|H-node}] [NBSCOpe=*string*]
 [NISDomain=*string*] [NIServers=*ipadd, ipadd...*]
 [NTPServers=*ipadd, ipadd...*]
 [POLICYfiltering=*ipadd, ipadd...*]
 [RESOURceserver=*ipadd, ipadd...*] [ROOTPath=*longstring*]
 [ROUTer=*ipadd, ipadd...*] [ROUTERDiscovery={ON|OFF}]
 [ROUTERSolicit=*ipadd*] [SERVER=*ipadd*]
 [SERVERName=*server-name*] [SOURCErouting={ENABLED|
 DISABLED}] [STATICroute=*ipadd, ipadd...*] [SUBLOCAL={ON|
 OFF}] [SUBNetmask=*ipadd*] [SWAPServer=*ipadd*]
 [T1TIme=*seconds*] [T2TIME=*seconds*] [TCPGarbage={ON|OFF}]
 [TCPKeepalive=*seconds*] [TCPTtl=*t1*]
 [TIMEOffset=*utc-offset*] [TIMEServer=*ipadd, ipadd...*]
 [TRAILerencap={ON|OFF}]
 [XDISplayservers=*ipadd, ipadd...*]
 [XFONTservers=*ipadd, ipadd...*]

where:

- *name* is a character string 1 to 15 characters long. It may contain any printable character.
- *seconds* is a time, time offset, or timeout value in seconds.
- *bootfilesize* is the length in 512-octet blocks of the default boot image for the client.
- *ipadd* is an IP address in dotted decimal notation.
- *string* is a character string 1 to 99 characters long. It may contain any printable character.
- *lease-time* is a time in seconds from 1 to 4294967295.
- *longstring* is a character string 1 to 254 characters long. It may contain any printable character.
- *t1* is a number from 1 and 255.
- *server-name* is a character string 1 to 63 characters long. It may contain any printable character.
- *utc-offset* is a time offset in seconds from Coordinated Universal Time (UTC).

Description This command modifies a standard pre-defined option in an existing DHCP policy. Pre-defined options are added using [add dhcp policy command on page 23-13](#). Each parameter has an associated IANA allocated option number which it defines, this is shown in brackets after the parameter description.

For more information about IANA allocated options, see <http://www.iana.org/assignments/bootp-dhcp-parameters>.

You can override any of these options by creating a user-defined option with the same number using the [add dhcp option command on page 23-11](#).

The **policy** parameter specifies the name of the policy in which the option is to be modified.

The **arptimeout** parameter specifies the timeout in seconds for ARP cache entries (Option 35).

The **bootfilesize** parameter specifies the length in 512-octet blocks of the default boot image for the client (Option 13).

The **broadcastaddress** parameter specifies the broadcast address in use on the client's subnet (Option 28).

The **cookieserver** parameter specifies a list of RFC 865 cookie servers available to the client. Cookie servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 8).

The **dnserver** parameter specifies a list of Domain Name System (RFC 1035) name servers available to the client. Domain Name System name servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 6).

The **domainname** parameter specifies the domain name that the client should use when resolving host names via the Domain Name System (Option 15).

The **etherencap** parameter specifies whether the client should use Ethernet Version 2 (RFC 894) or IEEE Standard 802.3 (RFC 1042) encapsulation for Ethernet interfaces (eth and vlan). A value of **off** indicates that the client should use RFC 894 encapsulation. A value of **on** means that the client should use RFC 1042 encapsulation (Option 36).

The **extensionpath** parameter specifies a string to specify a file, retrievable via TFTP, which contains information that can be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response (Option 18).

The **file** parameter specifies the boot file name for the client (Option 67).

The **hostname** parameter specifies the name of the client. The name may or may not be qualified with the local domain name. See RFC 1035 for character set restrictions (Option 12).

The **impresserver** parameter specifies a list of Imagen Impress servers available to the client. Imagen Impress servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 10).

The **intmtu** parameter specifies the interface's Maximum Transmission Unit (MTU), in bytes. This is the maximum packet size that the client can transmit over this interface at the physical layer (for example, Ethernet). Higher-layer packets (for example, IP datagrams) that are bigger than this is fragmented by the client and reassembled at the receiving end (Option 26).

The **ipforwarding** parameter specifies whether the client should configure its IP layer for packet forwarding. A value of **disable** disables IP forwarding, and a value of **enable** enables IP forwarding (Option 19).

The **ipmtu** parameter specifies the size, in bytes, of the largest IP packet the client should be prepared to reassemble from packets that were fragmented at the physical layer. The client discards IP packets that are larger than this value (Option 22).

The **ipplateau** parameter specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68 (Option 25).

The **iptimeout** parameter specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 (Option 24).

The **ipttl** parameter specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255 (Option 23).

The **logserver** parameter specifies a list of MIT-LCS UDP log servers available to the client. Log servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 7).

The **lprserver** parameter specifies a list of RFC 1179 line printer servers available to the client. Line printer servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 9).

The **maskdiscovery** parameter specifies whether the client should perform subnet mask discovery using ICMP. A value of OFF indicates that the client should not perform mask discovery. A value of ON means that the client should perform mask discovery (Option 29).

The **masksupplier** parameter specifies whether the client should respond to subnet mask requests using ICMP. A value of OFF indicates that the client should not respond. A value of ON means that the client should respond (Option 30).

The **meritdumpfile** parameter specifies the path name of a file where the client's core image should be dumped in the event the client crashes. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 14).

The **nameserver** parameter specifies a list of IEN116 name servers available to the client. IEN116 servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 5).

The **nbdbservers** parameter specifies a list of RFC 1001/1002 NetBIOS datagram distribution servers (NBDD) listed in order of preference. NetBIOS datagram distribution servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 45).

The **nbnameservers** parameter specifies a list of RFC 1001/1002 NetBIOS name servers (NBNS) listed in order of preference. NetBIOS name servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 44).

The **nbnodetype** parameter specifies the NetBIOS node type that allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002 (Option 46).

The **nbscope** parameter specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002 (Option 47).

The **nisdomain** parameter specifies the name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set (Option 40).

The **nisservers** parameter specifies a list of IP addresses indicating NIS servers available to the client. NIS servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 41).

The **nntpervers** parameter specifies a list of IP addresses indicating NTP servers available to the client. NTP servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 42).

The **policyfiltering** parameter specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks that specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next hop address does not match one of the filters should be discarded by the client. Policy filters are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list (Option 21).

The **resourceserver** parameter specifies a list of RFC 887 Resource Location servers available to the client. Resource Location servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 11).

The **rootpath** parameter specifies the path name that contains the client's root disk. The path name is formatted as a character string consisting of characters from the NVT ASCII character set (Option 17).

The **router** parameter specifies a list of IP addresses for routers on the client's subnet. Routers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Routers should be listed in order of preference (Option 3).

The **routerdiscovery** parameter specifies whether the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of **off** indicates that the client should not perform router discovery. A value of **on** means that the client should perform router discovery (Option 31).

The **routersolicit** parameter specifies the address where the client should transmit router solicitation requests (Option 32).

The **server** parameter specifies the address of the server to use in the next step of the client's bootstrap process. As the router is not capable of providing an operating system executable this option allows the IP address of an appropriate TFTP server to be set (Option 65).

The **servername** parameter specifies the name of the server host. This is passed to the client (Option 66).

The **sourcerouting** parameter specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of **disable** disallows forwarding of such datagrams, and a value of **enable** allows forwarding (Option 20).

The **staticroute** parameter specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. A maximum of up to 32 IP addresses can be specified in a comma separated list. The default route (0.0.0.0) is an illegal destination for a static route (Option 33).

The **sublocal** parameter specifies whether the client may assume that all subnets of the IP network where the client is connected use the same MTU as the subnet of that network where the client is directly connected. A value of **on** indicates that all subnets share the same MTU. A value of **off** means that the client should assume that some subnets of the directly connected network may have smaller MTUs (Option 27).

The **subnetmask** parameter specifies the client's subnet mask as defined in RFC 950. If you intend to use subnet or supernet addressing when you create DHCP ranges, you should include this option in the policy before you create the range (Option 1).

The **swapservers** parameter specifies the IP address of the client's swap server (Option 16).

The **t1time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **renewing** state (Option 58).

The **t2time** parameter specifies the time interval, in seconds, from address assignment until the client transitions to the **rebinding** state (Option 59).

The **tcpgarbage** parameter specifies whether the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of **off** indicates that a garbage octet should not be sent. A value of **on** indicates that a garbage octet should be sent (Option 39).

The **tcpkeepalive** parameter specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application (Option 38).

The **tcpttl** parameter specifies the default time-to-live value that the client should use when sending TCP segments (Option 37).

The **timeoffset** parameter specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC) (Option 2).

The **timeserver** parameter specifies a list of RFC 868 time servers available to the client. Time servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 4).

The **trailerencap** parameter specifies whether the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of **off** indicates that the client should not attempt to use trailers. A value of **on** means that the client should attempt to use trailers (Option 34).

The **xdisplayserver**s parameter specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. A maximum of up to 32 IP addresses can be specified in a comma separated list. Addresses should be listed in order of preference (Option 49).

The **xfontserver**s parameter specifies a list of X Window System Font servers available to the client. X Window System Font servers are specified by IP address. A maximum of up to 32 IP addresses can be specified in a comma separated list. Servers should be listed in order of preference (Option 48).

Examples To change the DN server for policy "base", use the command:

```
set dhcp poli=base dnss=192.168.100.51
```

Related Commands

- [add dhcp policy](#)
- [create dhcp policy](#)
- [delete dhcp policy](#)
- [destroy dhcp policy](#)
- [set dhcp option](#)
- [show dhcp policy](#)

set dhcp range

Syntax SET DHCP RANGE=*name* PROBE={ARP|ICMP}

where *name* is a character string 1 to 15 characters long. It may contain any printable character.

Description This command modifies the server's method for probing IP addresses in the specified range.

The **probe** parameter specifies how the DHCP server checks whether an IP address is being used by other hosts. If **arp** is specified, the server sends ARP requests to determine if an address is in use. If **icmp** is specified, the server sends ICMP Echo Requests (pings). The default is **icmp**.

Note that **arp** cannot be specified if the range includes a gateway (by specifying the **gateway** parameter when it was created), or if the network uses Proxy ARP.

Examples To set the range "office" to use ARP packets to probe IP addresses, use the command:

```
set dhcp ran=office pro=arp
```

Related Commands

- [add dhcp range](#)
- [create dhcp range](#)
- [delete dhcp range](#)
- [destroy dhcp range](#)
- [show dhcp range](#)

show bootp relay

Syntax SHow BOOTp RELAY

Description This command displays the current configuration of the BOOTP Relay Agent (Figure 23-1, Table 23-1).

Figure 23-1: Example output from the **show bootp relay** command

```

BOOTP Relaying Agent Configuration
-----

Status          : ENABLED
Maximum Hops    : 4

DHCP Option 82:
  Insertion status ..... Enabled
  Check ..... Yes
  Reforwarding policy ..... Replace
  Debugging ..... Disabled

BOOTP Relay Destinations
Address          Interface
-----
192.231.35.29   -
192.231.35.44   -
192.231.35.29   vlan1
192.231.35.44   vlan1
192.231.35.29   vlan2
-----

BOOTP Counters

InPackets ..... 4   OutPackets ..... 4
InRejects ..... 0
InRequests ..... 4
InReplies ..... 4
-----

```

Table 23-1: Parameters in output of the **show bootp relay** command

Parameter	Meaning
Status	Whether the BOOTP Relay Agent is enabled.
Maximum Hops	Maximum value allowed for the hops field in a BOOTP message before the message is discarded.
Insertion Status	The status of DHCP Option 82 insertion, either Enabled or Disabled.
Check	Whether DHCP Option 82 information returned from the DHCP server is being checked, either Yes or No.
Reforwarding policy	The re-forwarding policy for client DHCP packets that contain DHCP Option 82 information, either Replace, Keep, or Drop.
Debugging	The status of DHCP Option 82 debugging, either Enabled or Disabled.

Table 23-1: Parameters in output of the **show bootp relay** command (cont.)

Parameter	Meaning
BOOTP Relay Destinations	List of DHCP/BOOTP servers that BOOTREQUEST messages are forwarded to.
Address	IP address of the DHCP/BOOTP server.
Interface	Source interface of the DHCP/BOOTP clients for the server. A hyphen indicates that this is set to all interfaces on the router without an assigned relay destination.
InPackets	Total number of BOOTP packets received.
OutPackets	Total number of BOOTP packets transmitted.
InRejects	Number of incoming BOOTP packets rejected because of an error in the packet.
InRequests	Number of BOOTP requests received.
InReplies	Number of BOOTP replies received.

Related Commands

add bootp relay
delete bootp relay
disable bootp relay
enable bootp relay
purge bootp relay
set bootp relay maxhops
show bootp relay port

show bootp relay port

Syntax SHoW BOOTp RELAY Port [= {*port-list* | ALL}]

where *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

Description This command displays port-related information about the BOOTP Relay port settings (Figure 23-2, Table 23-2).

Use the **port** parameter to specify the port to display BOOTP Relay information for. If **all** is specified, information about all ports on the device is displayed.

Figure 23-2: Example output from the **show bootp relay port** command

```

BOOTP Relay Port Information:
-----
Port ..... 1
Trusted ..... No
Subscriber-ID ..... user12332

Port ..... 2
Trusted ..... Yes
Subscriber-ID .....
-----

```

Table 23-2: Parameters in output of the **show bootp relay port** command

Parameter	Meaning
Port	Number of the switch port.
Trusted	Whether the port is trusted.
Subscriber-ID	Subscriber-ID assigned to the port.

Related commands

- [disable bootp relay option82](#)
- [enable bootp relay option82](#)
- [set bootp relay option82](#)
- [set bootp relay option82 port](#)
- [show bootp relay](#)

show dhcp

Syntax SHow DHCP

Description This command displays the state of the DHCP module (Figure 23-3, Table 23-3).

Figure 23-3: Example output from the **show dhcp** command

```

DHCP Server

State ..... enabled
BOOTP Status ..... enabled
DEBUG Status ..... enabled
Extended Client ID . enabled
Policies ..... poll
                    prnt
Ranges .....,, develop (202.36.163.6 - 202.36.163.22)
                    remote (192.168.100.92 - 192.168.100.124)
In Messages ..... 3
Out Messages ..... 3
In DHCP Messages ... 3
Out DHCP Messages .. 3
In BOOTP Messages .. 0
Out BOOTP Messages . 0

DHCP Client

Interface ..... eth0
Client Identifier .. 00-00-cd-03-b3-4c-00-80-00-01
State ..... bound
Server ..... 10.194.0.10
Assigned Domain ....
Assigned IP ..... 10.194.0.1
Assigned Mask ..... 255.255.255.255
Assigned Gateway ... 0.0.0.0
Assigned DNS ..... 0.0.0.0

```

Table 23-3: Parameters in output of the **show dhcp** command

Parameter	Meaning
State	Whether the status of the DHCP server is enabled or disabled.
BOOTP Status	Whether the status of BOOTP serving is enabled or disabled.
Extended Client ID	Whether extended client IDs are transmitted by this device; either Enabled or Disabled.
BOOTP Status	Whether the status of BOOTP serving is enabled or disabled.
Policies	A list of the policies that have been defined.
Ranges	A list of the ranges that have been defined.
In Messages	Total number of DHCP or BOOTP messages received by the router.
Out Messages	Total number of DHCP or BOOTP messages transmitted by the router.
In DHCP Messages	Number of DHCP messages received by the router.
Out DHCP Messages	Number of DHCP messages transmitted by the router.

Table 23-3: Parameters in output of the **show dhcp** command (cont.)

Parameter	Meaning
In BOOTP Messages	Number of BOOTP messages received by the router.
Out BOOTP Messages	Number of BOOTP messages transmitted by the router.
Interface	Interface(s) on which this client is active.
Client Identifier	Identifying token used in DHCP messages for this client.
State	Current state of the DHCP client; either Renewing, Rebinding, Selecting, Requesting, Bound, or Init.
Server	DHCP server this client is connected to.
Assigned Domain	Domain name provided for this client by the DHCP server.
Assigned IP	IP address assigned to this client by the DHCP server.
Assigned Mask	IP address mask matching the address assigned to this client.
Assigned Gateway	Network gateway IP address provided by the DHCP server.
Assigned DNS	Domain Name Server IP address provided by the DHCP server.

Examples To display the current configuration of the DHCP server, use the command:

```
sh dhcp
```

Related Commands

- [disable dhcp](#)
- [enable dhcp](#)
- [set dhcp](#)
- [show dhcp client](#)
- [show dhcp policy](#)
- [show dhcp range](#)

show dhcp client

Syntax SHow DHCP CLIEnt [=ipaddress] [RANge=name] [PRObeconflicts]

Description This command displays information about the currently defined range client entries (Figure 23-4, Table 23-4 on page 23-54). If the **range** parameter is specified, then the clients in the specified range are displayed. If an IP address is specified on the **client** parameter, then information for that IP address is displayed.

If the **probeconflicts** parameter is specified, only clients that have replied to probes sent by the server to reclaim an IP address are displayed. This indicates a conflict between the IP addresses that these clients are currently using and the server's current client database.

Figure 23-4: Example output from the **show dhcp client** command

DHCP Client Entries					
IP Address	ClientId	State	Type	Expiry	
202.36.163.14	00-00-c0-00-00-01	unused	static	never	
202.36.163.15	00-00-c0-00-00-02	unused	static	never	
202.36.163.16	00-00-c0-00-00-03	unused	static	never	
202.36.163.17	00-00-c0-00-00-04	unused	static	never	
202.36.163.18	00-00-c0-00-00-05	unused	static	never	
202.36.163.19	00-00-c0-00-00-06	unused	static	never	
202.36.163.20	08-00-5a-a1-02-3f	inuse	auto	never	
202.36.163.21	00-00-c0-c9-c6-7b	inuse	auto	never	
202.36.163.22	08-00-09-0d-16-e7	inuse	auto	never	
202.36.163.23		unused	auto	never	
202.36.163.24		unused	auto	never	
202.36.163.25		unused	auto	never	
202.36.163.26		unused	auto	never	
202.36.163.27		unused	auto	never	
202.36.163.28	00-40-10-02-e8-a3	inuse	auto	never	
192.168.100.92	00-00-c0-c9-c6-21	inuse	dyn	19-Jun-1997 12:30:51	
192.168.100.93		unused	dyn		
192.168.100.94		unused	dyn		
192.168.100.95		unused	dyn		
192.168.100.96		unused	dyn		
192.168.100.97		unused	dyn		
192.168.100.98		unused	dyn		
192.168.100.99		unused	dyn		
192.168.100.110		unused	dyn		
192.168.100.111		unused	dyn		
192.168.100.112		unused	dyn		
192.168.100.113		unused	dyn		
192.168.100.114		unused	dyn		
192.168.100.115		reclaim	dyn		
192.168.100.116		reclaim	dyn		
192.168.100.117		reclaim	dyn		
192.168.100.118		reclaim	dyn		

Table 23-4: Parameters in output of the **show dhcp client** command

Parameter	Meaning
IP Address	IP address from the range of available IP addresses.
ClientId	Hardware address of the client, if any, that has been assigned the IP address.
State	State of the IP address: Unused - not currently in use and is available for assignment Inuse - currently assigned to a client Reclaim - currently being reclaimed
Type	Type of allocation mechanism applied to the IP address: Static - manual allocation Auto - automatic allocation Dyn - dynamic allocation
Expiry	Expiry date for dynamically allocated IP addresses.

Examples To display information about the clients in a range named “remote”, use the command:

```
sh dhcp clie ran=remote
```

Related Commands [show dhcp](#)
[show dhcp policy](#)
[show dhcp range](#)

show dhcp policy

Syntax SHow DHCP POLIcy[=*name*]

Description This command displays information about currently defined policies and the options configured for them. (Figure 23-5, Table 23-5 on page 23-56). If you specify a policy *name*, then information about that policy is displayed only.

Figure 23-5: Example output from the **show dhcp policy** command

```
DHCP Policies

Name: poll
  Base Policy: none
    01 subnetmask .... 255.255.255.0
    03 router ..... 202.36.163.21
    06 dnsserver ..... 192.168.100.50 192.168.100.33
    51 leasetime ..... 3600
*151 SVP server .... 192.168.88.20

Name: prnt
  Base Policy: poll
    01 subnetmask .... (poll) 255.255.255.0
    03 router ..... (poll) 202.36.163.21
    06 dnsserver ..... (poll) 192.168.100.50 192.168.100.33
    51 leasetime ..... (prnt) infinity
*151 SVP server .... (poll) (none)
*161 ..... (prnt) 192.168.4.2 192.168.6.2
*172 privservernum... (prnt) 4
*253 optionpresent... (prnt) (none)
*254 privservernam15. (prnt) privateserver
```

Table 23-5: Parameters in output of the **show dhcp policy** command

Parameter	Description
Name	The name of the policy.
Base Policy	The base policy inherited by this policy.
options...	<p>A list of the options configured for the specified policy.</p> <p>Each entry includes the following information:</p> <hr/> <p>The DHCP option identifier. This is the number that was assigned to the option.</p> <p>The number has an asterisk (*) on its left if its option is a user-defined option, configured using the add dhcp option command on page 23-11.</p> <hr/> <p>The parameter keyword. This is either:</p> <ul style="list-style-type: none"> the default assigned name for an option between the numbers of 1-68 that was configured using add dhcp policy, or a name that was user-defined for the option using add dhcp option. <hr/> <p>The current values of the option. If the option was configured using the add dhcp option command on page 23-11, then the option value is formatted based on the specified type.</p>

Examples To display information about the 'base' policy, use the command:

```
sh dhcp poli=base
```

Related Commands

- [add dhcp option](#)
- [add dhcp policy](#)
- [create dhcp policy](#)
- [delete dhcp option](#)
- [delete dhcp policy](#)
- [destroy dhcp policy](#)
- [set dhcp option](#)
- [set dhcp policy](#)
- [show dhcp](#)
- [show dhcp client](#)
- [show dhcp range](#)

show dhcp range

Syntax SHow DHCP RANge [=name]

where *name* is a character string 1 to 15 characters long. It may contain any printable character.

Description This command displays information about currently defined ranges (Figure 23-6, Table 23-6 on page 23-58). If a range name is specified, then information about it is displayed.

This command also displays counters for DHCP and BOOTP. BOOTP is used to transport DHCP messages. If the router is acting as a DHCP server for clients on subnets that are not directly connected to one of the router's interfaces, the DHCP messages are relayed through intermediate routers acting as BOOTP relay agents.

Figure 23-6: Example output from the **show dhcp range** command

```

DHCP Ranges

Name: remote
Policy ..... poll
Probe Type ..... ICMP
Start Address ..... 192.168.100.92
End Address ..... 192.168.100.124
Reclaim Status ..... Deferred
Next reclaim in ..... 5 seconds
Used Address(es) ..... 192.168.100.92      192.168.100.94      192.168.100.95
                          192.168.100.96
Free Address(es) ..... 192.168.100.93      192.168.100.97      192.168.100.98
                          192.168.100.99      192.168.100.100     192.168.100.101
                          192.168.100.102
Reclaiming Address(es) ..... 192.168.100.103      192.168.100.104      192.168.100.105
                          192.168.100.106      192.168.100.107      192.168.100.108
                          192.168.100.109      192.168.100.110      192.168.100.111
                          192.168.100.112      192.168.100.113      192.168.100.114
                          192.168.100.115      192.168.100.116      192.168.100.117
                          192.168.100.118      192.168.100.119      192.168.100.120
                          192.168.100.121      192.168.100.122      192.168.100.123

In DHCP Messages ..... 0
In Discover Messages ..... 0
In Request Messages ..... 0
In Decline Messages ..... 0
In Release Messages ..... 0
Out DHCP Messages ..... 0
Out Offer Messages ..... 0
Out Ack Messages ..... 0
Out Nak Messages ..... 0
In BOOTP Messages ..... 0
Out BOOTP Messages ..... 0

Total probe replies ..... 0
  Probe reply no details ... 0
  Probed entry static ..... 0
  Probed client new ..... 0
  Client has multiple IPs .. 0

```

Table 23-6: Parameters in output of the **show dhcp range** command

Parameter	Meaning
Name	Name of the range.
Policy	Policy that is applied to entries in the range.
Probe Type	Whether the DHCP server's method for probing IP addresses is ICMP or ARP.
Start Address	First IP address in the range.
End Address	Last IP address in the range.
Reclaim Status	Whether IP addresses are currently being reclaimed for clients: In progress Yes Stopped No Deferred No available route to the IP address being reclaimed
Next reclaim in	For ranges with deferred status, the time until the next reclaim is attempted.
Used Address(es)	List of IP addresses currently assigned to clients.
Free Address(es)	List of IP addresses currently available for assignment.
Reclaiming Address(es)	List of IP addresses currently being reclaimed from clients.
In DHCP Messages	Total number of DHCP messages the server received for this range.
In Discover Messages	Number of DHCP Discover messages the server received for this range. A client broadcasts these messages to initiate a DHCP session.
In Request Messages	Number of DHCP Request messages the server received for this range. A client uses this message to request parameters from a server that has offered them, to check that a previously allocated address is still correct (for example, after the client has rebooted) and to extend its lease of an address.
In Decline Messages	Number of DHCP Decline messages the server received for this range. When a client is offered an address, it may use ARP to check the address. The client sends a Decline message if it discovers that the IP address the server has offered it is already being used by another device.
In Release Messages	Number of DHCP Release messages the server received for this range. A client sends this message to relinquish an address when it no longer requires it. This makes the address available to another client.
Out DHCP Messages	Total number of DHCP messages the server transmitted for this range.
Out Offer Messages	Number of DHCP Offer messages the server sent for this range. The server sends this message in response to a client Discover message to offer configuration parameters to the client.
Out Ack Messages	Number of DHCP Acknowledgment messages the server sent for this range. The server sends this message in response to a client Request message, to supply the client with configuration parameters.
Out Nak Messages	Number of DHCP Negative Acknowledgment messages the server sent for this range. The server sends this message in response to a client Request message to tell the client that the IP address the client believes it has is wrong, the IP address the server offered is no longer available, or the client's lease has expired.

Table 23-6: Parameters in output of the **show dhcp range** command (cont.)

Parameter	Meaning
In BOOTP Messages	Number of BOOTP messages the server received for this range.
Out BOOTP Messages	Number of BOOTP messages the server transmitted for this range.
Total probe replies	The total number of probe replies received for IP addresses in this range. The server probes an IP address to determine if another host is using the address. The probe is either an ARP request or an ICMP Echo Request (ping). A probe reply indicates a conflict between the IP address that the client is currently using and the server's current client database.
Probe reply no details	The number of probe replies received where the server was unable to determine the client ID details of the responder. This results in the DHCP client entry appearing as 'inuse', but with no ClientId.
Probed entry static	The number of probe replies received when reclaiming static client entries after start-up.
Probed client new	The number of probe replies received from clients that are not already in the server's client database. This can indicate either that a statically configured client is setup within the DHCP range, or the DHCP file was deleted or out of date when the server was restarted. This can also occur if a client is moved from a dynamic lease to a static lease.
Client has multiple IPs	The number of times a client has replied to probes for more than one IP address. This can happen if the server is using ARP probes and a device in the network is configured for Proxy ARP.

Examples To display information about a range named “remote”, use the command:

```
sh dhcp rang=remote
```

Related Commands

- [add dhcp range](#)
- [create dhcp range](#)
- [delete dhcp range](#)
- [destroy dhcp range](#)
- [set dhcp range](#)
- [show dhcp](#)
- [show dhcp client](#)
- [show dhcp policy](#)

