

# Contents

The *Network Registrar CLI Reference Guide* is written for network and system administrators and is intended to provide information about how to use Cisco Network Registrar’s command line program, nrcmd.

Use this online guide after you have installed Network Registrar and have it running. This guide provides the following information:

- [Network Registrar CLI Introduction](#)  
 Provides instructions about how to use the nrcmd program, including batch and interactive operations, command syntax, command attribute guidelines, and navigation.
- [Network Registrar CLI Commands](#)  
 Gives a detailed description of all the nrcmd commands and their attributes, including usage guidelines and logging information.
- [Using the nrcmd Program in Scripts](#)  
 Provides suggestions about how to create batch files to execute nrcmd commands.
- [CLI Codes and Formats](#)  
 Describes status and error codes as well as dump and load formats.

This online guide uses the following notational conventions in command syntax:

Square brackets([ ])—Group optional elements.

Vertical bars (|)—Separate alternative mutually exclusive elements.

Angle brackets (< >)—Indicate you must provide a value for an attribute or argument in the command.

Copyright © 1992--2022 Cisco Systems, Inc. All rights reserved.

---

# Network Registrar CLI Introduction

[About the nrcmd Program](#)

[Invoking the nrcmd Command](#)

[Batch Mode](#)

[Interactive Mode](#)

[Environment Variables](#)

[Command Organization](#)

[Command Usage](#)

[Saving Your Changes](#)

[Refreshing and Clearing the CLI Cache](#)

[Navigation Keys](#)

## About the nrcmd Program

The **nrcmd** command line interface (CLI) enables you to configure and manage your DNS, DHCP, and TFTP servers. This section describes how to use the **nrcmd** CLI. It specifically describes:

- Invoking the command in batch and interactive modes
- Command organization and syntax
- Special keyboard navigation characters

## Invoking the nrcmd Command

You can invoke the **nrcmd** command in batch mode and run scripts that use the commands; or you can invoke the **nrcmd** command in interactive mode and enter commands at the **nrcmd** command prompt. By default, the **nrcmd** command is located in `/opt/nwreg2/local/usrbin` or `/opt/nwreg2/regional/usrbin`.

This method prompts for your user name and password. Alternatively, you can invoke the command from the command prompt using the following syntax:

```
nrcmd [general-options] [command] [options]
```

Table 1-1 describes the general options when invoking from the command prompt.

**Table 1-1 General Options to nrcmd Command**

Option	Description
-C <i>cluster</i>	Cluster ( <i>cluster</i> is the name of the machine on which the Network Registrar servers are running). If not specified, the cluster name defaults to localhost.
-N <i>user</i>	Network Registrar user name ( <i>user</i> ).
-P	

<i>password</i>	Network Registrar user password ( <i>password</i> ).
<b>-h</b>	Prints help text.
<b>-L</b>	Accesses the local cluster CLI.
<b>-R</b>	Accesses the regional cluster CLI.
<b>-b &lt; file.txt</b>	Batch file ( <i>file.txt</i> is the file of <b>nrcmd</b> commands that run in batch mode, read a line at a time and with a new line printed after the prompt).

---

**Note:** The cluster to which you connect determines the CLI attributes that appear and are available for the release of Network Registrar running on the cluster. This CLI Reference describes the attributes for the current release. For the attributes available for an earlier release, see the CLI Reference for that release.

---

### Batch Mode

The program goes into batch mode if you include a functional command or the **-b < file.txt** option on the line. The text file can include any number of **nrcmd** commands, and you can include comment lines preceded by the pound sign (#). In batch mode, you return to the normal system prompt. Note that display in batch mode is intended for parsing by an external program and, therefore, includes only command attributes that have values.

---

**Note:** The last line of code in the input file must end with an end-of-line character. It is also a good practice to make the last line of code an explicit **exit** command.

---

### Interactive Mode

The program goes into interactive mode if you enter just the **nrcmd** command, or include the *cluster*, *user*, or *password* options. To execute the CLI in interactive mode, enter:

```
nrcmd [-C cluster] [-N user] [-P password]
```

This syntax displays the interactive `nrcmd>` prompt, at which you enter a functional command and any optional parameters:

```
nrcmd> command [parameter parameter ...]  
system-response
```

To enter a series of attribute values, insert commas (,) between them. Do not add a space after the comma. If the value is a string containing one or more space characters, enclose the value in quotes:

```
nrcmd> zone example.com. set auth-servers=192.168.50.1,10.0.0.1  
100 Ok  
auth-servers=192.168.50.1,10.0.0.1
```

To terminate an interactive session, enter the **exit** command. To view the online help, enter the **help** command.

### Environment Variables

If you omit the general options, Network Registrar gets them from environment variables. If Network Registrar cannot find values for these parameters, it prompts you for them. If you omit the cluster name on a system where Network Registrar servers are installed, the **nrcmd** program assumes access to **localhost** and does not prompt you.

The environment variables that you can set that are recognized by the **nrcmd** program are CNR\_NAME for the name, CNR\_PASSWORD for the password, and CNR\_CLUSTER for the cluster name.

### Command Organization

The **nrcmd** commands specify a class of objects, which you can create, delete, or list. Each of these objects in turn has attributes, which you can enable, disable, set, get, and unset, depending on data type. These objects may also have common methods, which are specific to the type of object, and that let you perform operations on groups of attributes.

When you use the **nrcmd** commands to configure Network Registrar, you manipulate classes and command attributes.

## Classes

When you use the **nrcmd** commands to configure Network Registrar, you manipulate classes of objects, such as scopes, zones, and servers.

- **create**--Creates an entry. If the entry already exists, this command returns an error.
- **delete**--Removes an entry.
- **list**--Displays all the objects of a given type, including all attributes.
- **listnames**--Displays only the names of all objects of a given type.
- **show**--Displays the values of all the attributes.

## Attributes

- **enable**--Enables a Boolean type of attribute.
- **disable**--Disables a Boolean type of attribute.
- **set**--Sets the value of an attribute.
- **get**--Displays the value of an explicitly defined attribute.
- **unset**--Makes an attribute have no value. You cannot unset required attributes.

---

**Note** You cannot use **nrcmd** to **get** the value of implicitly defined attributes, including implicitly defined default values.

---

There are several ways to set attributes:

- **create** command. For example, to create a High-Availability (HA) DNS server pair, you can specify cluster and IP addresses for the main and backup servers during creation:

```
nrcmd> ha-dns-pair ha-pair-11 create 192.168.50.1 192.168.60.1 main=localhost  
backup=backup
```

- Use the **set** or **enable** command after creating the object. For example, you can set just the cluster references to the main and backup server for the created HA DNS pair:

```
nrcmd> ha-dns-pair ha-pair-1 set main=localhost backup=backup
```

- Add *attribute=value* pairs at the end of the **create** command.

If you use both the positional value and the *attribute=value* pair for the same attribute on the **create** command line, the *attribute=value* pair is the actual value used (because it comes last).

- Other custom methods--These are specific operations that you can perform on an object, beyond editing its attributes. Examples are adding a range of IP addresses to a scope, or removing hosts from a zone.

In interactive mode, all the attributes appear. In batch mode, only those attributes having values appear, and no default values appear. The display in batch mode is less user-friendly, but is more easily parsable by a program. These examples show how output compares in interactive and batch modes, respectively:

```
nrcmd> zone example.com show  
100 Ok  
example.com. (primary):  
  checkpoint-interval =  
  checkpoint-min-interval =
```

```
defttl = 12h
dynamic = [default=true]
dynupdate-set =
expire = 7d
...
```

```
$ nrcmd -N admin -P changeme zone example.com show
```

```
100 Ok
example.com.: defttl=12h; expire=7d; minttl=10m; nameservers={{0 rr2.example.com.}}; ns=rr2.; origin=example.com.; person=rr1.; refresh=3h;
retry=60m; serial=1; update-acl="key myKey";
100 Ok
```

## Command Usage

How you specify a series of arguments depends on the type of command you are using. The following subsections describe the differences between using the **create**, **set**, and **enable** commands.

### Create Keyword

When you use the **create** keyword and there are required arguments, you must supply them. You can also supply additional arguments. You must supply the required arguments in the specified order; however, you can specify the optional arguments in any order with the syntax `attribute=value`.

For example, the syntax for creating a scope is:

```
scope name create ipaddress mask [attribute=value]
```

This means that you must supply an IP address and mask when you create a scope, and you can optionally specify other attributes of the scope.

```
nrcmd> scope testScope create 192.168.50.0 255.255.255.0
100 Ok
testScope:
  addr = 192.168.50.0
  bootp = [default=disabled]
  deactivated =
```

You can also include attribute definitions on the same line. This example creates the same scope, but enables bootp support on it:

```
nrcmd> scope testScope create 192.168.50.0 255.255.255.0 bootp=enabled
100 Ok
testScope:
  addr = 192.168.50.0
  bootp = enabled
  deactivated =
```

After the **create** keyword creates and assigns all specified parameters to the object, it checks that all required attributes have values (either default or user-specified). If you omit required attributes, Network Registrar returns an error.

### Set Keyword

You use the set keyword to set the value of an attribute that is already created. If you want to set a list of values, such as DNS servers or IP addresses, you can separate them with commas.

This example specifies the name of the forward DNS zone to which a DHCP client's host name should be added for the testScope:

```
nrcmd> scope-policy testScope set forward-zone-name=example.com.
100 Ok
forward-zone-name=example.com.
```

This example specifies the list of IP addresses for zone transfers for a zone:

```
nrcmd> zone example.com. set auth-servers=192.168.50.1,10.0.0.1
100 Ok
auth-servers=192.168.50.1,10.0.0.1
```

This example sets a client's client-class and domain name:

```
nrcmd> client 00:d0:ba:d3:bd:3b set client-class-name=internal domain-name=example.com.
```

```
100 Ok
client-class-name=internal
domain-name=example.com.
```

The **unset** keyword places an attribute in the undefined state. The **get** keyword displays the value for an attribute.

## Enable Keyword

You use the **enable** keyword to enable a Boolean attribute. After you enable one Boolean attribute, you may need to set its associated attributes. Use the **disable** keyword to disable a Boolean attribute. You can use the **unset** keyword to remove the enabled or disabled state of the Boolean attribute.

This example enables incremental transfer processing for the DNS server:

```
nrcmd> dns enable ixfr-enable
100 Ok
ixfr-enable=enabled
```

Once incremental transfer is enabled, this example changes its expiration interval:

```
nrcmd> dns set ixfr-expire-interval=10d
100 Ok
ixfr-expire-interval=1w3d
```

---

**Note:** You cannot use **set** and **enable** on the same command line.

---

## Attribute Flags

Command are described as:

- Required--The attribute is required for the object, and usually syntactically positional on the **create** command line. You must set the attribute or accept its default, and you can modify the value. You cannot use the **unset** keyword to set a required attribute to undefined. Trying to do so returns the error message 386 - Required attribute cannot be deleted.
- Optional--The attribute is optional and does not require a value. You can set and reset the attribute, and you can use the **unset** keyword to make it undefined.
- Read-only--The attribute is immutable and read-only. You can use the **gett** keyword with the attribute, but you cannot set or unset it. Trying to set or unset a read-only attribute returns the error message 385 - Read-only attribute cannot be modified.

## Saving Your Changes

With new commands introduced in Network Registrar 6.2, **nrcmd** applies the changes you make immediately. (Commands introduced in Network Registrar 6.2 are listed in Release Notes). With the commands from prior releases, the CLI waits for one of these events to occur before it saves your changes to the database:

- Invoking the **save** command
- Exiting from **nrcmd**
- Reloading a server
- Adding a resource record or host to a zone

## Refreshing and Clearing the CLI Cache

The CLI caches many configuration objects that it reads. If multiple users are making changes simultaneously, one CLI instance might have cached an out of date version of an object. The **session cache refresh** command causes the CLI to clear its local cache of all unmodified objects, forcing it to reread objects from the configuration database. The **session cache clear** command forces the CLI to clear all cached data, whether or not unsaved changes were made.

## Navigation Keys

Table 1-2 lists keyboard navigation key combinations that are useful when entering **nrcmd** commands.

**Table 1-2 nrcmd Navigation Key Combinations**

<b>Key Combination</b>	<b>Action</b>
Control-a	Go to the beginning of the line
Control-b	Back one character (or the left arrow key)
Control-d	Delete one character
Control-e	Go to the end of the line
Control-f	Forward one character (or the right arrow key)
Control-k	Kill to the end of the line
Control-l	Redraw the line
Control-n	Next line in the history (or the down arrow key)
Control-p	Previous line in the history (or the up arrow key)
Control-t	Shift an individual character left
Control-u	Delete the line and move the cursor to the beginning of the line
Control-w	Delete one word backwards
Esc-b	Back one word
Esc-f	Forward one word



## Table of Contents

### Overview

[nrcmd](#)  
[intro](#)  
[expert](#)

### CLI Commands

[acl](#)  
[addr-trap](#)  
[address-block](#)  
[address-type](#)  
[admin](#)  
[auth-server](#)  
[auth-ad-server](#)  
[call-home](#)  
[ccm](#)  
[cdns](#)  
[cdns64](#)  
[cdnssec](#)  
[cdns-interface](#)  
[cdns-redirect](#)  
[cdns-firewall](#)  
[cdns-rate-limit](#)  
[cdns-forwarder](#)  
[cdns-exception](#)  
[certificate](#)  
[client](#)  
[client-class](#)  
[client-class-policy](#)  
[client-policy](#)  
[cluster](#)  
[debug](#)  
[dhcp](#)  
[dhcp-address-block](#)  
[dhcp-address-block-policy](#)  
[dhcp-dns-update](#)  
[dhcp-interface](#)  
[dhcp-listener](#)  
[dhcp-maintenance-window](#)  
[dhcp-subnet](#)  
[dns](#)  
[dns-enum-config](#)  
[dns-enum-domain](#)  
[dns-enum-number](#)  
[dns-interface](#)  
[dns-update-map](#)  
[dns-view](#)  
[dnssec](#)  
[dnssec-key](#)  
[exit](#)  
[export](#)  
[extension](#)  
[failover-pair](#)  
[group](#)  
[gss-tsig](#)  
[ha-dns-pair](#)  
[help](#)



[import](#)  
[key](#)  
[ldap](#)  
[lease](#)  
[lease-notification](#)  
[lease6](#)  
[license](#)  
[link](#)  
[link-policy](#)  
[link-template](#)  
[link-template-policy](#)  
[no](#)  
[option](#)  
[option-set](#)  
[owner](#)  
[prefix](#)  
[prefix-policy](#)  
[prefix-template](#)  
[prefix-template-policy](#)  
[policy](#)  
[region](#)  
[report](#)  
[reservation](#)  
[reservation6](#)  
[resource](#)  
[role](#)  
[router](#)  
[router-interface](#)  
[router-login-template](#)  
[router-type](#)  
[save](#)  
[scope](#)  
[scope-policy](#)  
[scope-template](#)  
[scope-template-policy](#)  
[server](#)  
[session](#)  
[show](#)  
[smart](#)  
[snmp](#)  
[snmp-interface](#)  
[subnet](#)  
[sync-from-dns](#)  
[task](#)  
[tenant](#)  
[tftp](#)  
[tftp-interface](#)  
[trap-recipient](#)  
[update-policy](#)  
[vpn](#)  
[zone](#)  
[zone-dist](#)  
[zone-template](#)

## nrcmd

nrcmd - run the Network Registrar command line interface

### Synopsis

```
nrcmd [flags] [<command>]
```

## Description

Flags can be zero or more of:

```
-C [<cluster>][:<port>] Specify the cluster to connect to and
                        optional port number.
-N <name>                Specify the administrator name.
-P <password>            Specify the administrator password.
-b                       Run in batch mode.
-R                       Connect to a regional cluster.
-v (or -vv)             Display the version (and PCV and build).
-V visibility            Specify the session visibility.
```

<cluster> can be a host name, IPv4 address, or IPv6 address enclosed in brackets (i.e., [2001:DB8::1]). Default is localhost.

In batch mode, input is performed a line at a time rather than a character at a time, and the prompt is terminated with a newline character rather than a space.

The visibility level controls which properties and features are visible to the session. A lower visibility level makes more properties visible. The default visibility level is 5, and should not be changed without guidance from Cisco technical support.

## Examples

```
% nrcmd -N admin -P changeme -C localhost
% nrcmd -N admin -P changeme -C [::1]:1234
```

## intro

intro - Introduction to nrcmd commands

## Synopsis

## Description

The nrcmd commands fall into two basic groups: regular and irregular. The regular commands manipulate configuration objects such as DHCP Scopes and DNS Zones in a standard fashion. The irregular commands do everything else that is useful. This page will describe the general pattern of the regular commands. The behavior of the irregular commands will be described in their individual man pages.

### Regular Command Form

Regular commands provide common functions for creating, deleting, viewing and editing objects of a given class.

#### Create

```
<cmd> <name> create [<required args>] [<prop>=<val>]
```

#### Delete

```
<cmd> <name> delete
```

#### List

```
<cmd> list
```

The list command lists full details on each object.

```
<cmd> listnames
```

The listnames command lists only the names each object.

```
<cmd> listbrief
```

The listbrief command lists brief details on each object (see the conf/nrcmd-listbrief-defaults.conf file for more details).

```
<cmd> listcsv
```

This listcsv command outputs the objects in CSV format.

Note that for most list operations, <prop>=<val> clauses may be specified to filter the list of returned objects to those where that property has that value. In addition, the following can be used:

For string, string (non-null terminated), case-insensitive strings, and named references, `<prop>=<val>` does a case-blind match whereas `<prop>=|<val>` does a case exact match. For IPv6 addresses and prefixes, `<prop>=<val>` does an exact match, whereas `<prop>=|<val>` will do a containment match (i.e., is the object's property value contained by the specified value). For flags, `<prop>=<val>` will match if the flag is set, whereas `<prop>=|<val>` will be an exact (binary value) match. For `<prop>=~<val>`, `<val>` is a regular expression and is used to match against the string representation of the object's property value. Note that specifying complex expressions may be difficult due to the limited character set possible for TCL strings.

There are also three special properties available:

- count-only may be specified to return only the count of the number of (matching) objects (no objects are printed).
- vpn=<val> can be specified for objects with a vpn-id property where <val> can be global, all, or a vpn name.
- view=<val> can be specified for objects with a view-id property where <val> can be default, all, or a view name.

And, `-format="<format-string>"` may be specified for listbrief to override the listbrief format definition for the command. See the `conf/nrcmd-listbrief-defaults.conf` file for more details on the syntax of the format-string.

For example, the following command will list all leases across all VPNs that are in the leased state:

```
lease listbrief -vpn=all state=leased
```

The listbrief operation can also be combined with the session log command (or copy & paste) and the CLI's batch processing capabilities to perform operations. For example, to force available all of the unavailable leases on the prefix named "testing", one could use:

```
session log force.txt
lease6 listbrief prefix-name=testing state=unavailable \
  -format="lease6 <ip6address> force-available"
session log
exit
nrcmd ... -b <force.txt
```

Note: The above filter syntax is experimental and subject to change.

#### Modify

`<cmd> <name> set <prop>=<value> [<prop>=<value> ...]`  
The set command takes two forms: 'set <prop> <value>' for setting a single property value, and 'set <prop>=<value> ...' for setting multiple property values in a single command.

For flag properties (AT\_FLAGSINT), `<prop>=+<value>` may be used to set the flag bits and `<prop>=-<value>` may be used to clear the flag bits.

Errors include:

- unknown property
  - if <prop> is not an property name for the object
- invalid format
  - if <value> is not in a valid format
- invalid value
  - if <value> is not semantically valid

`<cmd> <name> get <prop>`  
The get command returns the value of the named property.

`<cmd> <name> unset <prop> [<prop> ...]`  
The unset command makes the named properties have no value.

`<cmd> <name> enable <feature>`  
The enable command sets the value of the named feature to true.

`<cmd> <name> disable <feature>`  
The enable command sets the value of the named feature to false.

`<cmd> <name> show`  
The show command displays the value of the object.

#### Class Specific Commands (methods)

The configuration behavior of some objects may be enhanced by the addition of class specific commands to perform a useful action such as modifying complex properties, or controlling the objects behavior.

For example, DHCP Scope objects contain lists of address ranges from which leases may be offered. To manipulate this list of ranges, the scope command provides the commands: addRange, removeRange, and listRanges.

Another example is the force-available command provided by the lease command to tell the DHCP server that a given lease should be forced into the available state.

#### Regional Commands

When connected to a regional cluster, many object types support push, pull, and reclaim. Push distributes an object or all objects to a list of local clusters. Pull merges a local cluster object or all objects from its Replica data into the central configuration. Reclaim removes the object or all objects from a local cluster. Many of these commands require specifying one of the data synchronization modes:

- Ensure - Ensures that the local cluster has new data without affecting any existing data.
- Replace - Replaces data without affecting other objects unique to the local cluster.
- Exact - Available for all object operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster (for push) and regional (for pull).

#### Licensing

nrcmd requires the current cluster to have a valid license. If the license is invalid or has expired, only the 'license' command will be operational; it may be used to establish a new license key.

#### Return Codes

All nrcmd commands will return a status code as the first line of output. The status codes are heavily influenced by SMTP and other line oriented protocols. The first word of the line is a three digit status code, and the remaining words on the line are descriptive text that may or may not be constant for a given status code. The first digit of the status code determines the class of the status:

- 1xx - the command completed successfully, possibly with warnings
- 3xx there was some error in processing the command
- 4xx errors in communicating with the cluster database server
- 5xx there is was an internal error in the program

Note that it is unwise to check for only 100 as 101 and other return codes still mean the operation was generally successful.

#### Property Types

The properties that are manipulated by the set and get command have specific data types which determine the syntactically valid values. Some of the common property (or attribute) types are:

- AT\_STRING - a string, valid inputs are:
  - \* any text
  - Text strings with embedded spaces and characters such as brackets (these have special meaning for TCL) may need to be escaped when specified in a command.
  - For example:  
value=\  
"Optional\ items\ are\ in\ \[brackets]\\"
- AT\_INT - an integer, valid inputs are:
  - \* decimal digits, or
  - \* 0x followed by hex digits.
- AT\_BOOL - a boolean value, valid inputs are:
  - \* true, on, enabled, 1, or
  - \* false, off, disabled, 0.
- AT\_DATE - a date, valid inputs are:
  - \* 'forever'
  - \* +<integer time value>
  - \* a date/time string with the format  
"[ddd ]mmm dd hh:mm[:ss] yyyy"  
The string form may also need to be escaped when specified in a command.
  - For example:  
timestamp=\  
"Fri\ Jan\ 29\ 11:21:18\ 2016\\"
- AT\_TIME - a span of time, in seconds, valid inputs are:
  - \* decimal number of seconds
  - \* combination of numbers of weeks, days, hours, minutes, and seconds for example: 1w2d3h4m5s.
- AT\_IPADDR - an IP address, valid inputs are:
  - \* dotted quad format, for example: 10.24.1.2

AT\_SUBNET - a subnet, valid inputs are an IP address followed by a slash (/) and a subnet length, for example: 10.24.0.0/16.

AT\_IP6ADDR - an IPv6 address, valid inputs are:

- \* x:x:x:x:x:x:x:x, where the 'x's are one to four hexadecimal digits of the eight 16-bit pieces of the address, for example: 2001:DB8:0:0:8:800:200C:417A
- \* or its compressed form where one or more sets of runs of zeros is replaced by ::, for example: 2001:DB8::8:800:200C:417A

AT\_PREFIX - an IPv6 prefix, valid inputs are an IPv6 address followed by a slash (/) and a prefix length, for example: 2001:DB8::/32

AT\_IP6 - either an IPv6 address or IPv6 prefix (see above).

AT\_MACADDR - a MAC address, valid inputs are:

- \* raw hex digits, for example: 010203040506
- \* hex digits separated by ':', '.', or '-', for example: 01:02:03:04:05:06, 01-02-03-04-05-06, 01.02.03.04.05.06
- \* type and length, followed by hex digits, for example: 1,6,ab:01:cd:02:ef:03

AT\_RANGEINT - a range restricted integer

AT\_RANGETIME - a range restricted time value

AT\_ENUMINT - an enumerated integer

AT\_FLAGSINT - a bitmask with named bit positions

AT\_EXPR - Expressions (see the User Guide for more details). Note that expressions may be difficult to enter directly via nrcmd because of TCL character set limitations. Therefore, it is recommended to define the expression in a text file and then use the <attribute>=@<filename> syntax to set the expression from the contents of the file.

#### Validation

Data validation will be done at configuration creation and property modification time. The CLI will check for required valid values when a configuration object is created, and it will check the validity of property values when they are set.

Dangling references that are created by deleting a referred-to object, such as the policy for a scope, or the client class for a client will not be caught by the CLI.

## Examples

## Limitations

NRCMD commands are parsed using TCL and this can restrict the character set available for use, mostly for data values. Sometimes one can work around this limitation by using the \ before the special character, but this may not always work. It may be necessary to use the web UI or, in some cases, special support (such as for AT\_EXPR properties as mentioned above).

## expert

Expert mode commands

### Synopsis

```

ccm sync-from-dhcp [AddressSpace]
ccm sync-from-dns [ZoneData|Hosts]

ccm trimChangeSets <delete-age> [<db-max-records>]

ccm killConnection <id>

cdns execute dump-cache <filename>
cdns execute load-cache <filename>
cdns execute dump-reqlist <filename>

```

```

cdns execute flush-reqlist <filename>
cdns execute flush-ns-cache <ip-address>|all
cdns execute dump-ns-cache <filename>
cdns execute flush-negative

cluster <local-cluster> delete

dhcp setFailoverState <state>

object list -class=<classname>
object listbrief -class=<classname>
object listnames -class=<classname>
object <oid> [show] [-class=<classname>] [-db=<dbid>]
object <oid> delete [-class=<classname>] [-db=<dbid>] [-force]

server-agent <name> create [<attribute>=<value> ...]
server-agent <name> delete
server-agent list
server-agent listbrief
server-agent listnames
server-agent <name> show
server-agent <name> set <attribute>=<value>
                        [<attribute>=<value> ...]
server-agent <name> get <attribute>
server-agent <name> unset <attribute>

server-agent <name> enable <attribute>
server-agent <name> disable <attribute>

sync-from-dns

dns ha-sync-all-rrs main-to-backup | backup-to-main

zone <name> ha-sync-all-rrs main-to-backup | backup-to-main
zone <name> listRR

zone <name> signZone
zone <name> removeSignature

```

## Description

These commands are available only in expert mode and must be used with care. To enter expert mode, use:

```
nrcmd> session set visibility=3
```

The **ccm sync-from-dhcp** command can be used to synchronize CCM address space data from the DHCP server scope data.

The **ccm sync-from-dns** command can be used to synchronize CCM DNS zone and RR data or hosts from RR data from DNS. The sync-from-dns, which is retained for backwards compatibility, is the same as ccm sync-from-dns ZoneData.

The **ccm trimChangeSets** command can be used to initiate a trim of the change sets (change log) using the specified arguments.

**Warning:** This operation is usually NOT necessary and uses the values specified, which may be different than the periodic trimming done by CCM. This command should be used with extreme caution as it can delete data that should be retained.

Superuser admin can use **ccm killConnection** command to shutdown existing connection to CCM Server. Id should be same as returned by **ccm listConnections full** command.

The **cdns execute** command can only be run from the localhost that is running the cdns server. The commands supported are as follows:

dump-cache	dumps the in-memory cache to the specified file
load-cache	loads in-memory cache from a specified file
dump-reqlist	dumps the active query request list to the specified file
flush-reqlist	drops all active query requests
flush-ns-cache	drops information about all or a specific name server
dump-ns-cache	dumps the contents of the remote-ns-cache to the specified file
flush-negative	drops information about non-existing or empty domains

The local cluster can only be deleted while in expert mode.

The **dhcp setFailoverState** command can be used to force failover state changes. This must be used with extreme care and is not recommended.

The **object** command can be used to display (or delete) the object with a specified oid. If **-class=<classname>** is specified, the DB for that class is used (the classname must be specified using the correct case). If **-db=<dbid>** is specified, the specified DB is used (the dbid must be specified in uppercase). If neither **-class** or **-db** is specified, the CCM DB is assumed. If **-force** is specified for a delete, some checks and actions normally performed when deleting the object are bypassed.

The **object** command can also be used to list objects of a particular class; the **-class=<classname>** must be specified.

The **server-agent** command can be used to manipulate how the cnrservagt starts servers. Note that once changes are made, Network Registrar must be restarted before these changes will take effect.

Note: When setting server-agent attributes that contain TCL special characters (such as \$), create a file that contains the desired string and then set the attribute using:

```
server-agent <name> set <attribute>=@<file-name>
```

The **dns ha-sync-all-rrs** and **zone ha-sync-all-rrs** commands can be used to manually schedule HA zone sync for all zones, or a single zone, respectively. All RRs in the target zone will be overwritten by the source zone RRs. For example, specifying main-to-backup causes all RRs on the backup zone to be overwritten by RRs from the main zone.

**zone <name> listRR** command will include the RR order and weight preceding the RR information when executed in expert mode. The RR list will display each RR using the following format:

```
<order> <weight> <name> [<ttd>] <class> <type> <data>
```

**zone <name> signZone** command can be used to enable DNSSEC for the zone and add signatures for all RRs of the zone, when executed in expert mode.

**zone <name> removeSignature** command can be used to remove signatures for all RRs of the zone and disable DNSSEC for the zone, when executed in expert mode.

## Examples

## Limitations

# acl

acl - Manages DNS access control lists which are used to control zone access for DNS updates, zone transfers and queries

## Synopsis

```
acl list
acl listnames
acl <name> show
acl <name> create "<match-list>"
acl <name> delete
acl <name> get <attribute>
acl <name> set <attribute>=<value>
acl <name> unset <attribute>
acl <name> add "<match-list>"
acl <name> remove "<match-list>"

acl < <name> | all > pull < ensure | replace | exact > <cluster-name>
                        [-report-only | -report]
acl < <name> | all > push < ensure | replace | exact > <cluster-list>
                        [-report-only | -report]
acl <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The `acl` command is used to manage DNS ACLs which are used to restrict dynamic DNS updates, zone transfers and queries. Once you have created the `acl` object, it can be used with the `update-acl`, `restrict-xfer-acl` and `restrict-query-acl` on the DNS server or a zone object.

You can specify the `match-list` as a comma-separated list of values, enclosed in quotes, or you can use the `add` and `remove` commands to edit the match list. The `add` command will add elements to the end of the list. The `remove` command will remove the first matching element in the `match-list`.

Match list entries can consist of IP node or subnet addresses, TSIG keys, or ACLs. A TSIG key must also be preceded by the keyword `"key"`. The `"!"` notation can be used to negate an entry in the list.

Note: While match lists are displayed with spaces, spaces should not be specified in entered lists.

The `pull`, `push` and `reclaim` commands are only available when connected to a regional cluster. `Push` and `reclaim` allow a list of clusters or `"all"`.

## Examples

```
nrcmd> acl my-acl create "key my-key, 10.1.0.0/16"
nrcmd> acl my-acl set match-list="10.1.1.1/32,other-acl"
nrcmd> acl my-acl add "!10.2.0.0/16"
```

## Status

## See Also

[key](#)

## Attributes

### description [string](#)

An optional description for this ACL configuration.

### match-list [amelist](#)

Displays a comma-separated list of match elements, which can consist of IP node or subnet addresses, TSIG keys, or ACLs. You can also use the following reserved words as elements in a match list:

```
any
none
localhost
localnets
```

To specify more than one element in the match list, enclose the list in quotation marks; for example:

```
"192.168.2.1, localhost"
```

Use the object name to reference another ACL or TSIG key or GSS ACL to support Microsoft Secure updates.

A TSIG key must also be preceded by the keyword `key`; for example:

```
key mykey.
```

A GSS ACL must also be preceded by the keyword `grp`. To allow all Microsoft clients use the `any`, `domain-controller`, or `domain-computer` groups; for example:

```
grp any
grp domain-controller
grp domain-computer
```

You must specify subnet addresses in address/mask format. Use an exclamation point (!) to negate an entry in the list; for example:

```
"!192.168.3.0/24,!youracl"
```

Note: You can define the name reference to an ACL in the match list before you actually create the ACL. But the ACL must exist before you start or reload the DNS server. If the DNS server cannot resolve an ACL name reference on either the DNS server object or a zone object, it will flag the error and will not start.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## addr-trap

`addr-trap` - Configures free-address monitoring by the DHCP server

## Synopsis



```

addr-trap <name> create
addr-trap <name> delete
addr-trap enable <attribute>
addr-trap <name> disable <attribute>
addr-trap <name> set <attribute>=<value> [<attribute>=<value> ...]
addr-trap <name> unset <attribute>
addr-trap <name> get <attribute>
addr-trap list | listnames | listbrief
addr-trap <name> [show]

addr-trap < <name> | all > pull < ensure | replace | exact >
      <cluster-name> [-report-only | -report]
addr-trap < <name> | all > push < ensure | replace | exact >
      <cluster-list> [-report-only | -report]
addr-trap <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The addr-trap command configures values that the DHCP server uses to monitor free-address levels. Use this command with the SNMP server to provide SNMP notification trap messages as free-address levels change within the DHCP server.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

## Status

## See Also

[scope](#), [trap-recipient](#)

## Attributes

enable [bool](#) default = on

Indicates whether this configuration is active, causing scopes to track their free-address levels and possibly send trap events.

high-threshold [percent](#) default = 25%

Sets the level at which the 'low' threshold will be re-enabled, and the high-threshold trap will be generated. See the low-threshold for details on how the free-address level is calculated.

low-threshold [percent](#) default = 20%

Sets the free-address level at which a low-threshold trap will be generated, and the 'high' threshold will be re-enabled.

For scopes, the free-address level is calculated as follows:

```

100 * -----
      available non-reserved leases
      total configured leases

```

where the counts are the sum across all of the scopes included in the aggregation.

For prefixes, the free-address level is calculated as follows:

```

100 * -----
      max-leases - dynamic leases
      max-leases

```

where the counts are the sum across all of the prefixes included in the aggregation.

mode [enumint](#)(scope=1, network=2, selection-tags=3, prefix=4, link=5, v6-selection-tags=6, countonly=7, v6-countonly=8) default = scope

Indicates how scopes should aggregate their free address levels. The 'scope' mode causes each scope to track its own free-address level independently. This is an IPv4 only mode.

The 'network' mode causes all of the scopes configured by this object to aggregate their free-address levels if they share a 'primary-subnet'. This is an IPv4 only mode.

The 'selection-tags' grouping causes scopes to aggregate their free-address information together if they share a primary-subnet and if their lists of selection tags match exactly. This is an IPv4 only mode.

The 'prefix' mode causes each prefix to track its own free-address level independently. This is an IPv6 only mode.

The 'link' mode causes all of the prefixes configured by this object to aggregate their free-address levels if they share a link. This is an IPv6 only mode.

The 'v6-selection-tags' grouping causes prefixes to aggregate their free-address information if they share a link, and if their lists of selection tags match exactly. This is an IPv6 only mode. The 'countonly' and 'v6-countonly' groupings are used for the built-in aggregation objects that are used to provide top-utilized information when no traps are configured. No traps will be fired for these modes.

name [string](#) required,unique

Gives a unique name to the configuration object.

tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## address-block

address-block - Defines a contiguous block of IP address space

### Synopsis

```
address-block [<vpn-name>/]<address/mask> create
               [<attribute>=<value> ...]
address-block [<vpn-name>/]<address/mask> delete
address-block list
address-block listnames
address-block listbrief
address-block <name> show
address-block <name> get <attribute>
address-block <name> set <attribute>=<value> [<attribute>=<value> ...]
address-block <name> unset <attribute>

address-block <name> delegate <cluster-name>
address-block < <name> | all > push < ensure | replace | exact >
               <cluster-name> [-report-only | -report]
address-block <name> reclaim
```

### Description

An address block is an aggregate of IP addresses based on a power-of-two address space. For example, the 192.168.0.0/16 address block includes 65536 (2<sup>16</sup>) addresses.

Address blocks can be further divided into child address blocks and subnets. For example, you might want to divide the 192.168.0.0/16 address block further into four child address blocks: 192.168.0.0/17, 192.168.64.0/17, 192.168.128.0/17, and 192.168.192.0/17. A subnet is used to designate a leaf node of the address space that will not be further subdivided.

Address blocks are used as a management tool to group and report on address space usage. The owner and region properties on an address block or its parent can be used to constrain user access to address space reports.

The delegate, push, and reclaim commands are only available when connected to a regional cluster.

### Examples

```
nrcmd> address-block 192.168.0.0/16 create
```

### Status

### See Also

[subnet](#), [owner](#), [region](#)

### Attributes

address [subnet](#) required,immutable

Specifies the IP address and mask of the CCM address block, which was set at creation. This defines the address range

of the block. Use the set command to redefine the address.

**description** [string](#)

Describes how this address block is used.

**forward-zone-name** [dname](#)

Names the forward zone associated with this block.

**owner** [oid](#)

Names the owner of this address block. Use the owner field to group similarly owned address blocks; to limit administrative access; and to track allocation or delegation for ARIN reporting purposes.

**parent** [oid](#)

Identifies the parent address block.

**region** [oid](#)

Names the region associated with this address block. Use the region field to group similarly located address blocks and to limit administrative access.

**report-state** [enumint](#)(available=0, internal=1, reallocated=2, reassigned=3) transient

Transient report state of the CCM address block, provided to make it easier for ARIN reporting and for clients to filter the list of address blocks they display.

```
0 available
1 internal
2 reallocated
3 reassigned
```

**reverse-zone-name** [dname](#)

Names the reverse zone associated with this block.

**sink** [oid](#)

Points to either the CCMOwner or CCMCluster for a leaf block delegated to a lower-level sink. Delegated AddrBlocks (those with a non-null sink attribute) cannot be further split into child address blocks or subnets.

**source** [oid](#)

Points to the CCMOwner configuration object representing the source for a top-level block allocated from a higher-level source.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**type** [nameref](#)(0)

Defines an address-block type, enabling you to group address blocks that share common properties, such as their associated scope template. This attribute names a CCMAddrSpaceType object that contains more information about the type.

**vpn-id** [int](#) default = 0, immutable

Identifies the ID the VPN object used to support multiple address spaces, such as in a managed VPN environment.

## address-type

address-type - Configures a address space type

### Synopsis

```
address-type <name> create [<attribute>=<value>]
address-type <name> delete
address-type list
address-type listnames
address-type listbrief
address-type <name> show
address-type <name> set <attribute>=<value> [<attribute>=<value> ...]
address-type <name> get <attribute>
address-type <name> unset <attribute>

address-type <name> enable <attribute>
address-type <name> disable <attribute>
```

```
address-type < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
address-type < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
address-type <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The `address-type` command configures objects representing an address space type. These objects can be used to group similar address blocks or subnets that share common properties, such as their associated scope template.

The `pull`, `push`, and `reclaim` commands are only available when connected to a regional cluster. For `push` and `reclaim`, a list of clusters or "all" may be specified.

## Examples

## Status

## See Also

### Attributes

`client-class` [nameref](#)(0)

Identifies the client class associated with clients using this address space type.

`description` [string](#)

Provides a description of the address space type.

`name` [string](#) required,unique

Sets the name of the address space type.

`scopetemplate` [oid](#)

Defines a scope template to use when creating Scope objects from subnets allocated from address blocks of this type.

`selection-tags` [nlist](#)(obj(0))

Identifies the selection tags for Scopes of this address type.

`tenant-id` [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## admin

`admin` - Creates administrators and assigns them groups and passwords

## Synopsis

```
admin <name> create [<attribute>=<value>]
admin <name> delete
admin list
admin listnames
admin listbrief
admin <name> set <attribute>=<value> [<attribute>=<value> ...]
admin <name> get <attribute>
admin <name> unset <attribute>
admin <name> enable <attribute>
admin <name> disable <attribute>
admin <name> show
admin <name> enterPassword
admin <name> suspend
```

```

admin <name> reinstate

admin enterPassword

admin < <name> | all > pull < ensure | replace | exact > <cluster-name>
      [-report-only | -report]
admin < <name> | all > push < ensure | replace | exact > <cluster-list>
      [-omitrelated] [-report-only | -report]
admin <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The admin command configures administrators for the cluster.

You can choose any string for the name of the administrator. Names are not case-sensitive. Network Registrar uses a password to authenticate each administrator. Passwords are case-sensitive.

Because the password is sensitive information, Network Registrar prints its value as '\*\*\*\*\*'.

admin <name> enterPassword

If you want to enter a password and not have Network Registrar display the password on your screen, create an administrator but do not supply a password. Then use the enterPassword command to have Network Registrar prompt you twice for the password. If both entries match, Network Registrar will set the password value.

admin enterPassword

This is the only admin command available to admins without the ccm-admin subrole and can be used to change the current admin's password. It requires the admin to enter their current password and the desired new password (twice).

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified. For push, unless -omitrelated is specified, associated roles and groups are also pushed (using replace mode).

## Examples

## Status

## See Also

[group](#), [role](#)

## Attributes

**groups** [nlist](#)(obj(0))

Lists the names of groups to which this administrator belongs.

**password** [clrtxt](#)

Temporarily stores the cleartext password for this administrator that is used to create the password-secret. The maximum length is 255 characters.

**superuser** [bool](#)

Indicates whether this administrator is a superuser.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner associated with this admin.

**unlimited-sessions** [bool](#)

Indicates this administrator is permitted an unlimited number of concurrent token and user sessions.

auth-server - configures a External authentication server

## Synopsis

```
auth-server <name> create <address|ip6address>
    [<attribute>=<value> ...]
auth-server <name> delete
auth-server list
auth-server listnames
auth-server listbrief
auth-server <name> show
auth-server <name> get <attribute>
auth-server <name> set <attribute>=<value>
    [<attribute>=<value> ...]
auth-server <name> unset <attribute>

auth-server < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
auth-server < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
auth-server <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The auth-server command configures External Authentication servers.

For External Authentication using RADIUS feature:

1. Set the CCM auth-type attribute to Radius.
2. Configure one or more external auth-server objects.
3. Restart CPNR.

Note: Once enabled, if misconfigured, logins may fail. Refer to the local.superusers file information in the Administration Guide to be able to login and modify or disable the configuration.

If at least one external authentication server is enabled, external authentication will be used to authorize subsequent logins.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".

## Examples

## Status

## See Also

### Attributes

address [ipaddr](#)

Specifies the address for this remote authentication server.

ext-auth [bool](#) default = enabled

Enables this external authentication service.

ip6address [ip6addr](#)

Specifies the IPv6 address for this remote authentication server.

key [clrtxt](#)

Specifies the key used to communicate with this remote authentication server.

key-secret [secret](#)

Identifies the secret containing the key used to communicate with this remote authentication server.

name [string](#) required,unique

Identifies this remote authentication server.

port [rangeint](#)(1-65535) default = 1812

Specifies the port for this remote authentication server.

## auth-ad-server

auth-ad-server - Configures an external authentication active directory (AD) server

### Synopsis

```
auth-ad-server <name> create <addr> <domain>
    [<base-dn> <ad-group-name> <ad-user-attr-map>]
    [<attribute>=<value> ...]
auth-ad-server <name> delete
auth-ad-server list
auth-ad-server listnames
auth-ad-server listbrief
auth-ad-server <name> show
auth-ad-server <name> get <attribute>
auth-ad-server <name> set
    <attribute>=<value> [<attribute>=<value> ...]
auth-ad-server <name> unset <attribute>
auth-ad-server <name> addhost <host>
    [<addr>=<value> <port>=<value>]
auth-ad-server <name> listhost(s)
auth-ad-server <name> removehost <host>
    [<addr>=<value> <port>=<value>]

auth-ad-server < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
auth-ad-server < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
auth-ad-server <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The auth-ad-server command configures External Authentication Active Directory servers.

For auth-ad-server create command, the <addr> value format should be:

```
<host>:[<ip addr>@<port>],[<host>:[<ip addr>@<port>]],...
```

Where the <host> and <ip addr> are separated by a colon (:), the <ip addr> and <port> are separated using @, and the <host> is mandatory, and multiple objects can be separated by a comma (,).

The addhost command can be used to add a HostIPnPort object to an existing AD server object.

For External Authentication using AD feature:

1. Set the CCM auth-type attribute to Active-Directory.
2. Configure an AD server object with <addr> <domain> <base-dn> <ad-group-name> <ad-user-attr-map> attributes.
3. Restart CPNR.

Note: Once enabled, if misconfigured, logins may fail. Refer to the local.superusers file information in the Administration Guide to be able to login and modify or disable the configuration.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

### Examples

```
nrcmd> auth-ad-server authad create WIN-7NHA0SICRBC example.com
nrcmd> auth-ad-server authad create WIN-7NHA0SICRBC-1 example.com
    "CN=users" cpnr info
nrcmd> auth-ad-server authad addhost WIN-7NHA0SICRBC-2
    addr=10.106.200.207
nrcmd> auth-ad-server authad listhost
nrcmd> auth-ad-server authad removehost WIN-7NHA0SICRBC-2
```

### Status

## See Also

### Attributes

**ad-group-name** [string](#)

Specifies the Active Directory group name. Only users belonging to this group will be granted access to the Network Registrar application.

**ad-user-attr-map** [string](#)

Specifies the User object attribute to be used to get one or more Network Registrar groups.

**addr** [nlist\(obj\(0\)\)](#) required

Specifies the host name and Ports of the AD servers.

**base-dn** [string](#)

Specifies the distinguished name at which to start the search. User object will be searched in the base-dn and all entries in the tree below the base.

**domain** [rdname](#) required

Specifies the domain name to be used.

**name** [string](#) required,unique

Identifies this Active Directory remote authentication server.

**query-timeout** [time](#) default = 3s

Specifies the number of seconds the CCM server waits for a response to the search request. Default timeout is 3 seconds.

## call-home

call-home - Views and updates call-home configuration

### Synopsis

```
[no] call-home destination address http <url>
[no] call-home http-proxy <addr> port <port-number>
[no] call-home data-privacy hostname
[no] call-home debug <error|trace|detail|all>
```

```
show call-home detail
```

### Description

The **call-home** commands allows you to configure callhome transport for smart licensing on a regional cluster.

**call-home destination** command can be used to specify the URL for CSSM or satellite.

The **call-home debug** command allows you to set the call home debug level.

Any call-home configuration command, when executed with 'no' as prefix, unsets the specific configuration.

### Examples

### Status



## See Also

[smart](#), [license](#)

## ccm

ccm - Configures and controls the CCM server

### Synopsis

```
ccm get <attribute>
ccm set <attribute>=<value> [<attribute>=<value> ...]
ccm unset <attribute>
ccm show

ccm serverLogs show
ccm serverLogs nlogs=<nlogs> logsize=<logsize>

ccm listConnections [full]

ccm pullAddressSpace < update | complete | exact >
                        [-omitreservations] [-report-only | -report]
ccm pullIPv6AddressSpace < update | complete | exact >
                        [-report-only | -report]
ccm pullZoneData < update | complete | exact > [-omitrrs]
                        [-report-only | -report]
```

### Description

The **ccm** command manages the CCM server in the cluster.

The **ccm listConnections** command displays details on the current connections to the CCM server.

The **ccm pullAddressSpace**, **ccm pullIPv6AddressSpace**, and **ccm pullZoneData** commands are only available when connected to a regional cluster.

To export the change log, see the **export changeLog** command.

### Examples

### Status

## See Also

[export changeLog](#), [server](#)

### Attributes

**addrutil-poll-interval** [rangetime](#)(0-1y) default = 60m

Specifies how often to collect subnet and prefix utilization data from all the DHCP servers. If set to 0, polling is disabled.

**addrutil-poll-offset** [rangetime](#)(0-24h)

Provides a fixed time of day for subnet utilization polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided the polling interval is less than 24 hours, and the offset value is less than the polling interval. If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for polling will ensure that the first polling event occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

**addrutil-poll-retry** [rangeint](#)(0-4) default = 1

Specifies the number retries for a given polling interval, if polling fails.

**addrutil-trim-age** [rangetime](#)(24h-1y) default = 24w

Specifies the maximum age for subnet and prefix utilization data. Old data that exceeds this age will be trimmed at the next `addrutil-trim-interval`. Changes to this setting will take effect on the next server restart.

**addrutil-trim-interval** [rangetime](#)(0-1y) default = 24h

Specifies how often to trim old subnet and prefix utilization data. If set to 0 no automatic trimming occurs. Changes to this setting will take effect on the next server restart.

**admin-failed-login-limit** [rangeint](#)(0-10) default = 0

Specifies the maximum number of failed user or token login attempts that are allowed before an administrator account is suspended. If set to 0, no limit is applied. A value of 1 or 2 is not recommended. Changes to this setting will take effect the next time the server is restarted.

**admin-suspended-timeout** [rangetime](#)(0-2w) default = 24h

Specifies the length of time an administrator account should remain suspended if it has not been administratively reinstated. If set to 0, administrative action is required to reinstate the account. After this duration, the account will be automatically reinstated during next administrator login. Changes to this setting will take effect the next time the server is restarted.

**admin-token-session-limit** [rangeint](#)(0-100) default = 0

Specifies the maximum number of concurrent token sessions for a single administrator. Single sign-on connections are the most common token sessions. The web UI may also open token sessions for resource monitoring and dashboard displays. If set to 0, no limit is applied. A value of 1 or 2 may result in unexpected web UI failures and is not recommended. Changes to this setting will take effect the next time the server is restarted.

**admin-user-session-limit** [rangeint](#)(0-100) default = 0

Specifies the maximum number of concurrent user sessions for a single administrator. If set to 0, no limit is applied. Changes to this setting will take effect the next time the server is restarted.

**auth-type** [enumint](#)(Local=0, RADIUS=1, Active-Directory=2) default = Local

Sets the external authentication type to be used:  
0 Local  
1 RADIUS  
2 Active-Directory  
Changes to this setting will take effect the next time the product is restarted.

**idle-timeout** [time](#) default = 4h

Sets the maximum amount of time CCM will wait for a request over an incoming SCP connection. If set to 0, there is no idle timeout and CCM will wait forever. Also, see `unauth-idle-timeout`.

**is-rest-enabled** [bool](#) default = true

Controls whether the rest api is enabled. If the value is true, the rest api is enabled else disabled.

**lease-hist-compression** [bool](#) default = disabled

Specifies whether regional lease history records are written as compressed (enabled) or uncompressed (disabled) records. Compressing the records reduces the database size requirements, but slightly increases the processing requirements. Changes to this setting do not impact existing records. Both compressed and uncompressed records may co-exist in the same database. For optimum performance, it is recommended that the regional lease history database be compacted periodically, and especially after this feature is enabled. See the `cnrdb_util` utility to compact the databases. Note: Enabling this feature on an existing database will not result in the size of the database files decreasing, but may slow or prevent their growth. See above for compacting recommendations. Changes to this setting will take effect the next time the product is restarted.

**lease-hist-poll-interval** [rangetime](#)(0-1y) default = 4h

This interval specifies how often to collect the lease history information from all the DHCP servers. If set to 0, polling is disabled.

**lease-hist-poll-offset** [rangetime](#)(0-24h)

Provides a fixed time of day for lease history polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided the polling interval is less than 24 hours, and the offset value is less than the polling interval. If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for polling will ensure that the first polling event occurs at the offset time. For example, if you set the interval to

4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

#### **lease-hist-poll-retry** [rangeint](#)(0-4) default = 1

The number of retries for a given polling interval, if polling fails.

#### **lease-hist-trim-age** [time](#) default = 24w

The minimum length of time to retain lease history records in the lease history database. Any lease history record older than this time will be deleted when the next lease history database trimming operation occurs.  
If set to 0, no records will be trimmed.

#### **lease-hist-trim-interval** [rangetime](#)(0-1y) default = 24h

This interval specifies how often to trim the old lease history data. If set to 0 no automatic lease history trimming occurs. If lease history collection and polling are enabled and this parameter is set to 0, the lease history database will continue to grow without bound. Changes to this setting will take effect on the next server restart.

#### **lease-hist-txnseq-index** [bool](#) default = enabled

Specifies that CCM should maintain an index by transaction sequence number when polling DHCP lease history. This index is used with Java SDK methods to periodically export lease history data to an external process, server, or database. When disabled, these Java methods cannot be used to access lease history data. However, disk storage needed to maintain the lease history database is reduced. Changes to this setting will take effect on the next server restart, and apply only to new lease history records.

#### **local-edit-mode** [flags](#)(dhcp=2, dns=3) default = dhcp,dns

Indicates the default mode that web UI and CLI clients use for local edits:

- 2 dhcp  
If set, scope and reservation edits are forwarded to the DHCP server after being saved to the configuration database. If unset, a DHCP reload is required before the changes will take effect.
- 3 dns  
If set, zone and RR edits are forwarded to the DNS server after being saved to the configuration database. If unset, a 'Zone Distribution Sync' function is required to update the DNS server; a DNS server reload is also required for zone changes to take effect.

The default mode is applied only when the client requests the server default, or does not request a specific edit mode.

#### **log-settings** [flags](#)(database=1, authentication=2, scp-details=3, scheduled-tasks=4, licensing=5, server-events=6, replica=7, utilization=8, dnssec=9, lease-history=10, scp-request-detail=11, scp-response-detail=12) default = database,authentication,scp-details,scheduled-tasks,licensing,server-events,replica,utilization,dnssec,lease-history

Determines which events to log in the log files. By default, all but the scp-request-detail and scp-response-detail events are enabled.

Possible flags are:

- database  
This setting causes the server to log messages during local database processing.
- authentication  
This setting causes the server to log messages during user or session authentication.
- scp-details  
This setting causes the server to log messages when scp request is received, processed and scp response is sent out.
- scheduled-tasks  
This setting causes the server to log messages when tasks are scheduled or executed.
- licensing  
This setting causes the server to log messages when local cluster is registered and licensed with regional.
- server-events  
This setting causes the server to log messages during server event processing.
- replica  
This setting causes the server to log messages during replica database processing.
- utilization  
This setting causes the server to log messages during utilization database processing.
- dnssec  
This setting causes the server to log messages during dnssec key configuration, rollover processing and zone signing.
- lease-history  
This setting causes the server to log messages during lease history processing.
- scp-request-detail  
This setting causes the server to log a message about the details of received requests (see note below).
- scp-response-detail  
This setting causes the server to log a message about the details of responses to received requests (see note below.)

Note: The scp-request-detail and scp-response-detail settings are not recommended as they will reduce the performance, increase the logging load, and the data is a dump of the message object that requires product knowledge to decode. In general, scp-details should be sufficient for most situations.

Note: This attribute can be updated dynamically and does not require the CCM server reload.

#### **poll-replica-rrs** [enumint](#)(none=0, all=1, protected=2) default = 1

Specifies the server default setting when replicating RRs on primary zones. If set to none, no RR data will be replicated.

**poller-event-threads** [rangeint](#)(1-5) default = 2

This specifies how many threads that the poller will create.  
Changes to this setting will take effect on the next server restart.

**purge-userpreference-interval** [time](#) default = 6w

Specifies the length of time, after which, user preferences of an inactive administrator will be purged from DB. If set to 0, administrative action is required to clean up these. Changes to this setting will take effect the next time the server is restarted.

**regional-edit-mode** [flags](#)(admin=1, dhcp=2, dns=3) default =

Indicates the default mode that web UI and CLI clients use for regional edits:

- 1 admin  
When set, indicates that regional admin edits, including password changes made by individual users, will be automatically synchronized with all local clusters.
- 2 dhcp  
If set, reservation edits are forwarded to the local cluster or failover pair after being saved to the configuration database. If unset, push operations are required to update the local cluster(s).
- 3 dns  
If set, zone and RR edits are forwarded to the primary DNS server after being saved to the configuration database. If unset, a 'Zone Distribution Sync' function is required to update the DNS server.

The default mode is applied only when the client requests the server default, does not request a specific edit mode.

**replica-poll-interval** [ranetime](#)(0-1y) default = 4h

Specifies the server default value for how often to poll for configuration changes when replicating data from a local cluster.

**replica-poll-offset** [ranetime](#)(0-24h) default = 0

Provides a fixed time of day for replica polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided the polling interval is less than 24 hours, and the offset value is less than the polling interval. If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for polling will ensure that the first polling event occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

**subscription-mode** [bool](#) default = true

Controls whether the Software Innovation Access (SIA) license mode is enabled. This configuration is applicable for regional server running with Smart License.

**trim-changeset-age** [ranetime](#)(24h-3y) default = 24w

Sets the minimum age for a change log entry to be eligible for trimming.

**unauth-idle-timeout** [int](#) default = 300

Sets the maximum amount of time CCM will wait for a request over an incoming SCP connection before the user has been authenticated (see `idle-timeout` for the time used after a user has authenticated). If set to 0, the `idle-timeout` is used.

**webui-mode** [enumint](#)(Basic=0, Advanced=1, Expert=2) default = Basic

Sets the mode for the web UI at session startup:

- 0 basic
- 1 advanced
- 2 expert

## cdns

cdns - Configures and controls the DNS Caching server

### Synopsis

```
cdns disable <attribute>
cdns enable <attribute>
```

```
cdns get <attribute>
cdns set <attribute>=<value> [<attribute>=<value> ...]
cdns unset <attribute>
cdns show
```

```
cdns addForwarder <domain> [tls=on|off] [tls-auth-name=<name>] <addr>
cdns removeForwarder <domain> [<addr> ...]
cdns listForwarders
```

```

cdns addException <domain> [prime=on|off] [views=on|off] [tls=on|off] [tls-auth-name=<name>]
[<addr> ...]
cdns removeException <domain> [<addr> ...]
cdns listExceptions

cdns addRootHint <name> <addr> [<addr> ...]
cdns removeRootHint <name>
cdns listRootHints

cdns flushCache [<domain>]

cdns flushName <name> [<type>|*] [<view>] [<dns64>]

cdns serverLogs show
cdns serverLogs nlogs=<nlogs> logsize=<logsize>

cdns getStats [ <server | top-names | rate-limit | all> [total | sample] ]
cdns resetStats

```

## Description

The `cdns` command lets you configure the DNS Caching server in the cluster.

```

cdns addForwarder <domain> [tls=on|off] [tls-auth-name=<name>] <addr>
cdns removeForwarder <domain> [<addr> ...]
cdns listForwarders

```

Use the `Forwarder` commands to specify the addresses of any DNS servers that you want your Network Registrar DNS Caching server to use as forwarders for a specified domain. Network Registrar DNS Caching server forwards recursive queries to these servers and the receiving server should be prepared to resolve the query to completion. Typically forwarders are other DNS Caching servers that have access to internet or external DNS resources.

The `addForwarder` command adds the address of a forwarding server for the specified domain. `<name>` is the domain that this forwarder will apply to, and `<addr>` can be an ipv4 or ipv6 address followed by an optional port number (i.e. `<addr>[@<port>]`) or the name of a server (it must be possible to resolve the server name before it is used). If the `tls` flag is on, the server will connect to the name server using `tls`. By default, these flag are off. If `tls-auth-name` is provided server will verify this name in the `tls` certificate provided by name server.

The `removeForwarder` command removes the forwarder.

The `listForwarders` command lists the forwarders for this DNS server.

```

cdns addException <domain> [prime=on|off] [views=on|off] [tls=on|off] [tls-auth-name=<name>]
[<addr> ...]
cdns removeException <domain> [<addr> ...]
cdns listExceptions

```

Use the `exception` commands only if you do not want your DNS Caching server to use the standard name resolution for querying root name servers for names outside the domain. Network Registrar sends non-recursive queries to these servers and the receiving server should be authoritative for the exception domain. Typically exceptions are used to access local DNS authoritative resources (i.e. a company's internal zones).

The `addException` command lets you specify the resolution exception domains and the IP addresses of the associated servers. The addresses can be ipv4 or ipv6 with an optional port number (i.e. `<addr>[@<port>]`) or the name of a server (it must be possible to resolve the server name before it is used). If the `prime` flag is on, the server will query the name server for an updated list of name servers for the domain. If the `views` flag is on, the server will forward view information to the name server. The server will send non-recursive queries to the exception servers. This command may also be used to change the `prime` and `views` flags of an existing exception. By default, these flags are off. If the `tls` flag is on, the server will connect to the name server using `tls`. By default, these flag are off. If `tls-auth-name` is provided server will verify this name in the `tls` certificate provided by name server.

The `removeException` command removes an entry for exceptional resolution of addresses within a domain.

The `listExceptions` command lists the domains that are configured to have exceptional resolution of their names.

NOTE: If you have a forwarder and exception for the exact same domain, the DNS Caching server will use the forwarder for queries on that domain rather than the exception.

**cdns addRootHint** <name> <addr> [<addr> ...]

**cdns removeRootHint** <name>

**cdns listRootHints**

Use the `RootHint` commands to add or remove the names and addresses of the root servers. After you specify the root servers, Network Registrar queries them for their root name server records. These records are in turn used to resolve other names. As such, these values need not be exact, but must be accurate enough for the Network Registrar DNS Caching server to retrieve the correct information.

The `addRootHint` command adds the name of a root server and the root name server address(es). Addresses can be either `ipv4` or `ipv6` with an optional port number (i.e. `<addr>[@<port>]`).

The `removeRootHint` command removes a root server from the list.

The `listRootHints` command lists the root server information.

**cdns flushCache** [<domain>]

The `flushCache` command deletes cached RRs at or below the specified domain. If no domain is provided, deletes all RRs from the cache.

**cdns flushName** <name> [<type>|\*] [<view>] [<dns64>]

The `flushName` command will delete RRs from the cache with the given name and optional type in the specified view and `dns64`. If no type or `*` is provided, removes RRs of type `A`, `AAAA`, `PTR`, `CNAME`, `MX`, `SOA`, `NS`, `NAPTR`, `SRV` and `DNAME`. If no view or `dns64` is specified, the default view and no `dns64` is used.

**cdns serverLogs show**

**cdns serverLogs nlogs=<nlogs> logsize=<logsize>**

The `serverLogs show` command displays the number of log files and the maximum size for each file.

The `serverLogs` command allows setting the two server logging parameters, `nlogs` and `logsize`. Either or both may be specified in the command, and changes will only occur to the one(s) specified. When setting `logsize`, a suffix of `K` or `M` indicates units of thousands or millions.

```
cdns serverLogs nlogs=6 logsize=500K
cdns serverLogs logsize=5M
```

Note: For these changes to take effect you must save the changes and restart the server Agent.

**cdns getStats** [<statistics-category> [**total** | **sample**]]

The `getStats` command displays the requested DNS Caching server statistics, either since the last reload or for the last sample period. One of the following categories may be supplied: `all`, `server`, `top-names`, or `rate-limit`.

**cdns resetStats**

The `resetStats` commands returns the DNS Caching server activity counters (statistics) to zero.

## Examples

## Status

## See Also

[server](#)

## Attributes

`acl-blocklist` [amelist](#)

Sets the access block list for the server. Packets from clients on this

list will be ignored.

#### **acl-do-not-query** [amelist](#)

Specifies IP addresses or subnets that may not be queried. Can be IP4 or IP6. Append /num to indicate a classless delegation netblock, for example like 10.2.3.4/24 or 2001::11/64.

#### **acl-query** [amelist](#) default = any

Sets the access control lists for the server.

#### **activity-summary-interval** [rangetime](#)(60s-24h) default = 60s

Specifies the interval at which to log activity-summary information. Note that activity-summary must be specified in the log settings in order for this interval to take effect.

#### **activity-summary-settings** [flags](#)(query=1, query-type=2, cache=3, resol-queue=4, responses=5, memory=6, firewall=8, top-names=9, system=10, rate-limiting=11, security=12, upstream=13) default = query,cache,resol-queue,memory,system,security

Determines the category of statistics that is logged as part of activity summary. Note, activity-summary must be specified in the log-settings in order for this setting to take effect. The possible settings are:

- cache  
Logs statistics on the RR cache.
- firewall  
Logs statistics on DNS Firewall usage.
- memory  
Logs statistics on memory usage.
- query  
Logs statistics related to incoming queries.
- query-type  
Logs statistics on the RR types that are being queried.
- resol-queue  
Logs statistics on the resolution queue.
- responses  
Logs statistics about query responses.
- security  
Logs statistics related to security events.
- system  
Logs statistics on system usage.
- top-names  
Logs the top names queried and hit count.
- rate-limiting  
Logs the number of rate limiting events.
- upstream  
Logs the numbers of upstream queries.

#### **activity-summary-type** [flags](#)(sample=1, total=2) default = sample

Determines whether the CDNS server logs sample and/or total statistics when it logs activity-summary information. Note, activity-summary must be specified in the log-settings in order for this setting to take effect.

#### **allow-any-query-acl** [amelist](#) default = none

Allows ANY query requests from clients listed in this access control list. This list can contain hosts, network addresses and/or other ACLs. Request from clients matching this acl will be allowed ANY query.

#### **cache-max-negative-ttl** [time](#) default = 60m

Limits the time negative responses are stored in the cache beyond the limit specified in its SOA record in the authority section.

#### **cache-max-ttl** [time](#) default = 24h

Sets the time to live maximum for RRsets and messages in the cache. If the maximum kicks in, responses to clients still get decrementing TTLs based on the original (larger) values. When the internal TTL expires, the cache item has expired. Can be set lower to force the resolver to query for data often, and not trust (very large) TTL values.

#### **cache-min-ttl** [time](#) default = 0

Sets the time to live minimum for RRsets and messages in the cache. If the minimum kicks in, the data is cached for longer than the domain owner intended, and thus less queries are made to look up the data. Zero makes sure the data in the cache is as the domain owner intended, higher values, especially more than an hour or so, can lead to trouble as the data in the cache does not match up with the actual data any more.

#### **contrack-usage-warning** [percent](#) default = 50%

Specifies the percentage of the firewall connection tracking usage that will trigger a warning. Warnings are logged at the activity-summary-interval whenever the number of connections is at or above the specified percentage. Setting the value to 0 disables the warnings.  
Note: This attribute is only applicable to Linux systems.

It is highly recommended that connection tracking not be used for DNS related activity as this can block legitimate traffic

(incoming or outgoing) and result in resolution failures for clients or other DNS servers.

It is also strongly recommended that the firewall be completely disabled when using the DNS server as this can cause significant performance impact to DNS services especially for high query loads.

#### **hardened-glue** [bool](#) default = on

Specified if glue should only be trusted if it is within the servers authority.

#### **https** [bool](#) default = disabled

Enables or disables HTTPS support for CDNS. DoH is supported only for incoming queries. DoH supports GET and POST methods as specified in RFC8484. Before enabling HTTPS, the private and public key files must be placed in the CDNS data directory under `cdns/tls` and the `service-key` and `service-pem` attributes set. If using managed CDNS certificates, the certificate settings will be automatically set. Otherwise, the public certificate file must be placed in the CDNS data directory under `cdns/tls` and the `service-pem` attribute set.

#### **https-port** [rangeint](#)(1-65535) default = 443

The port number on which to provide HTTPS service. The CDNS server will not serve non-https queries on this port.

#### **immediate-response-stats** [bool](#) default = disabled

Enables collecting response time statistics when queries are replied to immediately. If this feature is disabled, the related statistics will show zero.

#### **listen-ip-version** [flags](#)(ipv4=1, ipv6=2) default = ipv4,ipv6

Controls which ip packets to accept and issue, IPv4, IPv6, or both.

#### **listen-protocol** [flags](#)(udp=1, tcp=2) default = udp,tcp

Controls which packet protocol to answer and issue, UDP, TCP, or both.

#### **log-settings** [flags](#)(config=1, server-ops=2, server-detailed-ops=3, scp=4, activity-summary=5, query=6, name-servers=7) default = config,server-ops,activity-summary

Determines which detailed events the Caching DNS server logs, as set using a bit mask. Logging these additional details can help analyze a problem. Leaving detailed logging enable for a long period, however, can fill the log files and cause the loss of important information. The possible flags are:

- `config`  
Controls logging pertaining to server configuration and server de-initialization (unconfiguration).
- `scp`  
Controls logging pertaining to SCP Message processing.
- `server-detailed-ops`  
Controls logging of detailed logging of server operations.
- `server-ops`  
Controls logging of high level logging of server operations.
- `activity-summary`  
This setting will cause a summary message to appear at an interval specified by `activity-summary-interval`. The summary provides detailed statistics about the servers operation.
- `query`  
This setting will cause logging of all DNS queries to the server. Note that query log entries are written to a separate `cdns_query_log` file.
- `name-servers`  
This setting enables logging when name servers for exceptions and forwarders become unresponsive or again become responsive.

#### **minimal-responses** [bool](#) default = true

Controls whether the DNS Caching server omits records from the authority and additional sections of query responses when these records are not required. When enabled (default) this may improve query performance, reduce query response size and avoid TCP fallback.

#### **msg-cache-size** [int](#) default = 1073741824

Sets the size of the message cache in bytes.

#### **neg-cache-size** [int](#) default = 10485760

Sets the size of the aggressive negative cache in bytes.

#### **packet-log-settings** [flags](#)(query-in=1, query-out=2, response-in=3, response-out=4) default = query-in,response-out

Determines the type of packets to log when packet logging is enabled. Packet logging can be enabled by configuring



the packet-logging attribute.

**query-in**  
This setting enables logging of incoming query packets. These are packets coming in from DNS clients.

**query-out**  
This setting enables logging of outgoing query packets. These are queries going to upstream DNS servers.

**response-in**  
This setting enables logging of incoming query response packets. These are responses coming from upstream DNS servers.

**response-out**  
This setting enables logging of outgoing query response packets. These are responses going to DNS clients.

#### **packet-logging** [enumint](#)(disabled=0, summary=1, detail=2) default = disabled

Determines the type packet logging that is logged to the CDNS logs. The type packets logged can be controlled with the packet-log-settings attribute.

**disabled**  
This settings disables packet logging.

**summary**  
This setting enables one line summary packet logging.

**detail**  
This setting enables detailed packet tracing.  
NOTE: This setting may significantly increase the amount of information that is logged and should only be used on a temporary basis for debugging purposes.

Note that while packet logging can be helpful for debugging and troubleshooting, it does have an impact on DNS server performance. Therefore, Cisco does not recommend leaving packet logging enabled in production environments.

#### **packet-logging-file** [enumint](#)(none=0, cdns=1, packet=2, all=3) default = packet

Determines the destination log of packet log messages when packet logging is enabled.

**cdns**  
Packet logging messages are logged to the standard CDNS log file (cdns\_log\*).

**packet**  
Packet logging messages are logged to a separate CDNS packet log file (cdns\_query\_log\*).

#### **port** [rangeint](#)(1-65535) default = 53

Specifies the UDP and TCP port number that the DNS caching server uses to listen for queries.

#### **prefetch** [bool](#) default = off

Sets whether message cache elements should be prefetched before they expire to keep the cache up to date. Turning it on gives about 10 percent more traffic and load on the machine, but popular items do not expire from the cache.  
When prefetch is enabled, records are assigned a prefetch time that is within 10 percent of the expiration time. As the server processes client queries and looks up the records, it checks the prefetch time. Once the record is within 10 percent of its expiration, the server will issue a query for the record in order to keep it from expiring.

#### **randomize-query-case** [bool](#) default = disabled

Enables the use of 0x20-encoded random bits in the query to foil spoof attempts. This perturbs the lowercase and uppercase of query names sent to authority servers and checks if the reply still has the correct casing.

#### **randomize-query-case-exclusion** [nlist](#)(obj(0))

Allows to create an exclusion list for randomization of upstream queries. This attribute will be used when randomize-query-case is enabled.

#### **remote-ns-cache-numhosts** [int](#) default = 100000

Sets the number of hosts for which information is cached.

#### **remote-ns-host-ttl** [time](#) default = 15m

Sets the time to live (ttl) for entries in the remote name server cache. The remote name server cache contains roundtrip timing (rtt), lameness and EDNS support information. Once an entry expires, it is removed from the remote name server cache and the next time the server is contacted a new entry will be added.

Note that the rtt is used to decide which name server to query. If a timeout occurs, the rtt value of that server is doubled. If a server starts to become unresponsive, a probing scheme is applied in which a few queries are selected to probe the IP address. If that fails, the name server is blocked for 15 minutes (remote-ns-host-ttl) and re-probed with one query after that. Therefore, it may be necessary to decrease the remote-ns-host-ttl in order to allow probing more frequently.  
The remote name server cache is not flushed after a CDNS server reload, but can be flushed using the nrcmd command

cdns execute flush-ns-cache.

**round-robin** [bool](#) default = true

Specifies whether you want round-robin cycling of equivalent records in responses to queries. Equivalent records are records of the same name and type. Since clients often only look at the first record of a set, enabling this features can help balance loads and keep clients from forever trying to talk to an out-of-service host.

**rrset-cache-size** [int](#) default = 1073741824

Sets the size of the RRset cache in bytes.

**security-event-log-settings** [flags](#)(configuration=1, dnssec=2, packet-inspection=3, rate-limit=4, cisco-umbrella=5) default = configuration,dnssec,packet-inspection,rate-limit,cisco-umbrella

Specifies the DNS security events that should be logged. When the CDNS server detects a security event and the related security event log setting is enabled, a log message will be written to the cdns security log file. In order for this setting to take effect, the log-settings attribute must include the security setting. Note that security-event-logging and security-event-log-settings configuration changes take effect immediately without requiring a CDNS server reload.

**cisco-umbrella**  
A security event log message will be generated when Cisco Umbrella forwarders respond with redirected addresses. Note that Cisco Umbrella forwarders must be configured in order for this security event to be caught. Note that a Cisco Umbrella subscription may be required.

**configuration**  
A security event log message will be generated based on DNS server configuration settings (i.e. ACL failures).

**dnssec**  
A security event log message will be generated is the CDNS server fails to validate DNSSEC data. DNSSEC validation failures may an indicate a cache poisoning attempt.

**packet-inspection**  
A security event log message will be generated based on DNS server detecting issues in the request packet. These issues may be detected by basic packet inspection (i.e. packet-inspection setting) or during packet processing. Excessive malformed packets may indicate a DoS attack.

**rate-limit**  
A security event log message will be generated if the CDNS server reaches configured IP and/or domain rate limits. Excessive DNS traffic requiring rate limiting may indicate an amplification attack.

**security-event-logging** [enumint](#)(disabled=0, enabled=1) default = disabled

Enables DNS security event logging based on settings configured in security-event-log-settings. Security event log messages are written to the cdns security log file. Note that security-event-logging and security-event-log-settings configuration changes take effect immediately without requiring a CDNS server reload.

**security-event-max-qname-size** [rangeint](#)(0-255) default = 0

Specifies the maximum size of a query name (QNAME) allowed. If a longer hostname is detected, the server will trigger a packet inspection DNS security event for the DNS tunneling category and the query will be refused. A setting of 0 (default) disables query name length checking.

**service-key** [string](#)

Defines the file name which contains the private key to be used by DNS for TLS and DoH sessions. The file must be in the CDNS data directory under the tls subdirectory (i.e. /cdns/tls). The openssl tool can be used to create TLS private and public key files.

**service-pem** [string](#)

Defines the pem file name which contains the public key certificate to be used by CDNS for TLS and DoH sessions. The file must be in the CDNS data directory under the tls subdirectory (i.e. /cdns/tls). Note that if using managed CDNS certificates, this attribute will be ignored and should be left unset.

**smart-cache** [bool](#) default = disabled

Specifies if the Caching DNS Server should use Smart Caching. When smart-cache is enabled, the Caching DNS server continues to use it's last best known answer when cached responses have expired and it cannot reach the authoritative name servers. The RRs in smart cache responses will have a 0 TTL. Smart Caching is useful to mitigate network outages and possible DDoS attacks that make the authoritative name servers unavailable. Enabling smart-cache automatically enables prefetch.

**smart-cache-expiration** [time](#) default = 0

When smart-cache is enabled, specifies a time limit for responding with expired RRs.

The default is 0, which allows the server to respond with expired answers as long as they remain in the cache.

**smart-cache-expiration-reset** [bool](#) default = disabled

When smart-cache is enabled and smart-cache-expiration is greater than 0, will reset the expiration time on active queries. This allows active queries to return expired answers, while allowing others to return SERVFAIL responses for a short period. Once the queries become active, will return expired answers. Default is disabled.

**smart-cache-expired-reply-ttl** [int](#) default = 30

When smart-cache is enabled, specifies the TTL value to use when replying with expired data.

**tls** [bool](#) default = disabled

Enables or disables TLS support for CDNS. Before enabling TLS, the private key files must be placed in the CDNS data directory under cdns/tls and the service-key attribute set. If using managed CDNS certificates, the certificate settings will be automatically set. Otherwise, the public certificate file must be placed in the CDNS data directory under cdns/tls and the service-pem attribute set. Enabling or disabling TLS service requires a CPNR service restart for the change to take effect.

**tls-port** [rangeint](#)(1-65535) default = 853

The port number on which to provide TCP TLS service. The CDNS server will not serve non-TLS queries on this port.

**tls-upstream-cert-bundle** [string](#)

Defines the file name which contain the certificate bundle. These certificates are used for TLS connections made to outside peers. These certificates are used to authenticate connections made to upstream DNS servers. The file must be in the CDNS data directory under the tls subdirectory (i.e. /cdns/tls).

**top-names** [bool](#) default = enabled

Specifies whether to collect top names data. When enabled, a snapshot of the cache hits for the top names that are queried is collected for each interval set by the top-names-max-age value. The list of top names that is reported with activity summary statistics is the most current snapshot.

**top-names-max-age** [rangetime](#)(10s-5m) default = 60s

Specifies the maximum age (based on last access time) of a queried name allowed in the list of top names.

**top-names-max-count** [rangeint](#)(5-1000) default = 10

Specifies the maximum number of entries in the list of top names queried. This limit is applied to the lists of top names that are logged or returned as part of activity summary.

**traps-enabled** [flags](#)(all=1, server-start=2, server-stop=3) default =

Determines the traps that this server is configured to send.

- 1 all  
Sends notifications for all server events.
- 2 server-start  
Sends notifications whenever the server is started or reinitialized.
- 3 server-stop  
Sends notifications whenever the server is stopped.

## cdns64

cdns64 - Controls and configures DNS64 processing in the DNS Caching server

### Synopsis

```
cdns64 <name> create <attribute>=<value>
cdns64 <name> delete
cdns64 <name> show
cdns64 <name> get <attribute>
cdns64 <name> set <attribute>=<value>
cdns64 <name> unset <attribute>
cdns64 list
cdns64 listnames
cdns64 listbrief
```

```
cdns64 < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
cdns64 < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
cdns64 <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The `cdns64` command lets you create and edit DNS64 objects. The `dns64` attribute would be enable by-default on DNS64 object creation. The DNS Caching servers must be reloaded for changes to take effect.

The `pull`, `push` and `reclaim` commands are only available when connected to a regional cluster. `Push` and `reclaim` allow a list of clusters or "all".

## Examples

```
nrcmd> cdns64 dns64 create
nrcmd> cdns64 dns64 set acl-match-clients=baaa::56ff:febd:3d6
nrcmd> cdns64 dns64 set prefix=64:ff9b::/96
nrcmd> cdns64 dns64 show
nrcmd> cdns64 dns64 delete
```

## Status

## See Also

### Attributes

**acl-match-clients** [amelist](#) default = any

Specifies the ACL that maps clients to this DNS64 object based on the client's source address.

**description** [string](#)

An optional description for this DNS64 configuration.

**dns64** [bool](#) default = true

Determines whether or not to enable DNS64 processing. DNS64 synthesizes AAAA records from A records, when a client queries for AAAA records, but none are found.

**name** [string](#) required,unique

The name of the DNS64 configuration instance.

**prefix** [prefix](#) default = 64:ff9b::/96

Specifies the IPv6 prefix to use for synthesizing AAAA records. The prefix length must be 32, 40, 48, 56, 64, or 96, and bits 64-71 of the prefix must be zero.

**priority** [priority](#)

Specifies the DNS64 priority relative to other DNS64 objects. The lowest non-zero value has the highest priority, zero priority object will have least priority. Each DNS64 should be assigned a unique priority to ensure consistent behavior.

If unset when added, the next non-zero value will be assigned.

## cdnssec

`cdnssec` - Controls and configures DNSSEC processing in the DNS Caching server

## Synopsis

```
cdnssec create <attribute>=<value>
cdnssec delete
```

```
cdnssec show
cdnssec get <attribute>
cdnssec set <attribute>=<value>
cdnssec unset <attribute>
```

## Description

The `cdnssec` command is used to control and manage DNSSEC processing in the DNS Caching server. Creating the `cdnssec` object does not enable DNSSEC in the server. In order to enable DNSSEC, the `dnssec` attribute must be explicitly enabled and the DNS Caching server must be reloaded.

## Examples

```
nrcmd> cdnssec create
nrcmd> cdnssec enable dnssec
nrcmd> cdnssec set trust-anchor-file=example.com.anchor
```

## Status

## See Also

### Attributes

**auto-trust-anchor-file** [nlist](#)(obj(0)) default = root.anchor

Defines files with a trust anchor for one zone each, which is tracked with RFC5011 probes. The probes are several times per month, thus the machine must be online frequently. The initial file can be one with contents as described in `trust-anchor-file`. The file is written to when the anchor is updated, so the server must have write permission. The files must be in the `data/cdns` directory.

**dnssec** [bool](#) default = disabled

Enables validation of DNS information using DNSSEC.

**domain-insecure** [nlist](#)(obj(0))

Defines domain names to be insecure, DNSSEC chain of trust is ignored towards the domain names. So a trust anchor above the domain name can not make the domain secure with a DS record, such a DS record is then ignored. Also keys from DLV are ignored for the domain. If you set trust anchors for the domain they override this setting (and the domain is secured). This can be useful if you want to make sure a trust anchor for external lookups does not affect an (unsigned) internal domain. A DS record externally can create validation failures for that internal domain.

**key-cache-size** [int](#) default = 4194304

Sets the size of the key cache in bytes.

**prefetch-key** [bool](#) default = off

Sets whether the DNS caching server should fetch the DNSKEYs earlier in the validation process, when a DS record is encountered. This lowers the latency of requests. It does use a little more CPU. Also if the cache is set to 0, it is no use.

**trust-anchor-file** [nlist](#)(obj(0))

Defines a file with trusted keys for validation. Both DS and DNSKEY entries can appear in the file. The format of the file is the standard DNS Zone file format. Default is no trust anchor file. The files must be in the `data/cdns` directory.

## cdns-interface

`cdns-interface` - Configures the DNS Caching server's network interfaces

## Synopsis

```

cdns-interface <name> create [<attribute>=<value>]
cdns-interface <name> delete
cdns-interface list
cdns-interface listnames
cdns-interface listbrief
cdns-interface <name> show
cdns-interface <name> set <attribute>=<value> [<attribute>=<value> ...]
cdns-interface <name> get <attribute>
cdns-interface <name> unset

cdns-interface <name> enable <attribute>
cdns-interface <name> disable <attribute>

```

## Description

The `cdns-interface` command configures network interfaces for use by the Network Registrar DNS Caching server. If there are no defined interfaces, the server discovers and uses all available interfaces on the system. When this list is present, the server uses only the available interfaces, if any, that match this list.

## Examples

## Status

## See Also

### Attributes

`address` [subnet](#)

Specifies the IP address and subnet mask of the DNS interface.

`ip6address` [prefix](#)

Specifies the IPv6 address and prefix length for one or more DNS interfaces.

`name` [string](#) required,unique

Specifies the user-assigned name of the DNS caching server interface.

`port` [rangeint](#)(1-65535)

Specifies the UDP and TCP port number the DNS server listens on. If no port is specified, will use the port configured on the Caching DNS Server.

## cdns-redirect

`cdns-redirect` - Controls and configures DNS redirect processing in the DNS Caching server

## Synopsis

```

cdns-redirect <name> create <attribute>=<value>
cdns-redirect <name> delete
cdns-redirect <name> show
cdns-redirect <name> get <attribute>
cdns-redirect <name> set <attribute>=<value>
cdns-redirect <name> unset <attribute>
cdns-redirect list [-priority]
cdns-redirect listbrief [-priority]
cdns-redirect listnames [-priority]

cdns-redirect < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
cdns-redirect < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
cdns-redirect <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The `cdns-redirect` command is used to control and manage DNS redirect processing in the DNS Caching server. The DNS Caching server must be reloaded for changes to take effect. The `-priority` option sorts lists by the priority instead of the name, which is the default.

The `pull`, `push` and `reclaim` commands are only available when connected to a regional cluster. `Push` and `reclaim` allow a list of clusters or "all".

## Examples

```
nrcmd> cdns-redirect HandleMissing create
nrcmd> cdns-redirect HandleMissing set action=nxdomain aaaa-response=64:ff9b::10
```

## Status

## See Also

### Attributes

#### `a-response` [ipaddr](#)

Specifies the IPv4 address to use as a response to a query for an A resource record. Only applies to `redirect` and `redirect-nxdomain` actions.

#### `aaaa-response` [ip6addr](#)

Specifies the IPv6 address to use as a response to a query for a AAAA resource record. Only applies to `redirect` and `redirect-nxdomain` actions.

#### `action` [enumint](#)(`refuse=1, redirect=2, drop=4, redirect-nxdomain=5, rpz=6`) default = `drop`

Specifies what kind of action to take when this rule is executed.

- `drop` - Ignore/drop this query completely.
- `refuse` - Respond with no data and the REFUSED status.
- `redirect` - Redirect A/AAAA queries to the specified IP address.
- `redirect-nxdomain` - Redirect A/AAAA queries if the query domain does not exist.
- `rpz` - Use Response Policy Zone (RPZ) rules.

#### `client-acl` [amelist](#) default = `any`

Lists the clients that will be redirected by this rule.

#### `description` [string](#)

An optional description for this DNS firewall rule.

#### `domains` [nlist](#)(`obj(0)`)

Lists the domains that will be redirected by this rule. Sub-domains of the listed domains are also redirected. If the list is empty, all domains will match this rule. Does not apply to the `nxdomain` action.

#### `name` [tag](#) required,unique

Specifies the name of a redirect rule. The allowed characters include `a-z`, `A-Z`, `0-9`, `.`, `-`, `_`, and `+`.

#### `priority` [priority](#) default = `0`

Specifies the rule priority relative to other rules. The lowest non-zero value has the highest priority. A priority of `0` (the default) has the lowest possible priority.

#### `rpz-override` [enumint](#)(`none=0, drop=1, no-op=2, nodata=3, nxdomain=4, redirect=5`) default = `none`

Specifies an override clause for responses matching RPZ rules. Overrides cause the response to differ from that stated in the RPZ zone.

- `none` - No override (default).
- `drop` - Overrides all RPZ rule matches by dropping the request.
- `no-op` - Overrides all RPZ rule matches by following normal request processing (allow listing RPZ rules).
- `nodata` - Overrides all RPZ rule matches to return NODATA.

```
nxdomain - Overrides all RPZ rule matches to return NXDOMAIN.
redirect - Overrides all RPZ rule matches to be redirected
           to FQDN specified in rpz-override-redirect.
```

#### rpz-override-redirect [dname](#)

Specifies an FQDN to use in conjunction with rpz-override when the redirect override is selected. All RPZ rule matches will be redirected to this FQDN.

#### rpz-server-addr [nlist\(obj\(0\)\)](#)

Specifies a list of one or more IP addresses (v4 and/or v6) of the Response Policy Zone Authoritative DNS server(s). This attribute only applies to rpz actions.

#### rpz-tls [bool](#) default = false

Enables TLS transport for this RPZ. If you enable this, also configure a tls-cert-bundle to load CA certs, otherwise the connections cannot be authenticated. Default is disabled.

#### rpz-tls-auth-name [string](#)

Defines the auth name for the RPZ server. If TLS is enabled, the cdns server will check the TLS authentication certificates with that name sent by the RPZ server.

#### rpz-trigger [flags\(query-name=1, ip-answers=2, ns-name=3, ns-ip=4\)](#) default = query-name

Specifies which part of query processing should apply RPZ rules.

- query-name - Apply RPZ rules to incoming query FQDN.
- ip-answers - Apply RPZ rules to A/AAAA records in answer section (rpz-ip).
- ns-name - Apply RPZ rules to name server names used in query processing (rpz-nsdname).
- ns-ip - Apply RPZ rules to name server IPs used in query processing (rpz-nsip).

#### rpz-zone-name [dname](#)

Specifies the zone name to use for Response Policy Zone rule processing. Only applies to rpz actions.

## cdns-firewall

cdns-firewall - Controls and configures DNS firewall processing in the DNS Caching server

### Synopsis

```
cdns-firewall <name> create <attribute>=<value>
cdns-firewall <name> delete
cdns-firewall <name> show
cdns-firewall <name> get <attribute>
cdns-firewall <name> set <attribute>=<value>
cdns-firewall <name> unset <attribute>
cdns-firewall list
cdns-firewall listbrief
cdns-firewall listnames
```

### Description

The cdns-firewall command is used to control and manage DNS firewall processing in the DNS Caching server. The DNS Caching server must be reloaded for changes to take effect.

### Examples

```
nrcmd> cdns-firewall HandleMissing create
nrcmd> cdns-firewall HandleMissing set action=drop aaaa-response=64:ff9b::10
```

### Status

### See Also

### Attributes



**a-response** [ipaddr](#)

Specifies the IPv4 address to use as a response to a query for an A resource record. Only applies to redirect and redirect-nxdomain actions.

**aaaa-response** [ip6addr](#)

Specifies the IPv6 address to use as a response to a query for a AAAA resource record. Only applies to redirect and redirect-nxdomain actions.

**action** [enumint](#)(refuse=1, redirect=2, drop=4, redirect-nxdomain=5, rpz=6) default = drop

Specifies what kind of action to take when this rule is executed.

- drop - Ignore/drop this query completely.
- refuse - Respond with no data and the REFUSED status.
- redirect - Redirect A/AAAA queries to the specified IP address.
- redirect-nxdomain - Redirect A/AAAA queries if the query domain does not exist.
- rpz - Use Response Policy Zone (RPZ) rules.

**client-acl** [amelist](#) default = any

Lists the clients that will be redirected by this rule.

**description** [string](#)

An optional description for this DNS firewall rule.

**domains** [nlist](#)(obj(0))

Lists the domains that will be redirected by this rule. Sub-domains of the listed domains are also redirected. If the list is empty, all domains will match this rule. Does not apply to the nxdomain action.

**name** [tag](#) required,unique

Specifies the name of a redirect rule. The allowed characters include a-z, A-Z, 0-9, ., -, \_, and +.

**priority** [priority](#) default = 0

Specifies the rule priority relative to other rules. The lowest non-zero value has the highest priority. A priority of 0 (the default) has the lowest possible priority.

**rpz-override** [enumint](#)(none=0, drop=1, no-op=2, nodata=3, nxdomain=4, redirect=5) default = none

Specifies an override clause for responses matching RPZ rules. Overrides cause the response to differ from that stated in the RPZ zone.

- none - No override (default).
- drop - Overrides all RPZ rule matches by dropping the request.
- no-op - Overrides all RPZ rule matches by following normal request processing (allow listing RPZ rules).
- nodata - Overrides all RPZ rule matches to return NODATA.
- nxdomain - Overrides all RPZ rule matches to return NXDOMAIN.
- redirect - Overrides all RPZ rule matches to be redirected to FQDN specified in rpz-override-redirect.

**rpz-override-redirect** [dname](#)

Specifies an FQDN to use in conjunction with rpz-override when the redirect override is selected. All RPZ rule matches will be redirected to this FQDN.

**rpz-server-addr**s [nlist](#)(obj(0))

Specifies a list of one or more IP addresses (v4 and/or v6) of the Response Policy Zone Authoritative DNS server(s). This attribute only applies to rpz actions.

**rpz-tls** [bool](#) default = false

Enables TLS transport for this RPZ. If you enable this, also configure a `tls-cert-bundle` to load CA certs, otherwise the connections cannot be authenticated. Default is disabled.

**rpz-tls-auth-name** [string](#)

Defines the auth name for the RPZ server. If TLS is enabled, the `cdns` server will check the TLS authentication certificates with that name sent by the RPZ server.

**rpz-trigger** [flags](#)(query-name=1, ip-answers=2, ns-name=3, ns-ip=4) default = query-name

Specifies which part of query processing should apply RPZ rules.

- query-name - Apply RPZ rules to incoming query FQDN.
- ip-answers - Apply RPZ rules to A/AAAA records in answer section (rpz-ip).
- ns-name - Apply RPZ rules to name server names used in query processing (rpz-nsdname).
- ns-ip - Apply RPZ rules to name server IPs used in query processing (rpz-nsip).

rpz-zone-name [dname](#)

Specifies the zone name to use for Response Policy Zone rule processing. Only applies to rpz actions.

## cdns-rate-limit

cdns-rate-limit - Controls and configures DNS rate limiting in the DNS Caching server

### Synopsis

```
cdns-rate-limit show

cdns-rate-limit get <attribute>
cdns-rate-limit set <attribute>=<value>
cdns-rate-limit unset <attribute>
cdns-rate-limit enable <attribute>
cdns-rate-limit disable <attribute>

cdns-rate-limit add [domain=<domain>
                    [[applies-to=]domain|subdomain|both]
                    [[rate-limit=<rate-limit>]
cdns-rate-limit mod[ify] <index> [domain=<domain>]
                               [applies-to=domain|subdomain|both]
                               [rate-limit=<rate-limit>]
cdns-rate-limit rem[ove] <index>
cdns-rate-limit list
```

### Description

The `cdns-rate-limit` command is used to control and manage DNS rate limiting in the DNS Caching server. The DNS Caching server must be reloaded for changes to take effect.

The `get`, `set`, `unset`, `enable` and `disable` subcommands affect the general settings of this object.

The `add`, `modify`, `remove` and `list` subcommands manipulate the domain-rate-limiting-list attribute. The `<index>` refers to an entry's position in the list.

Note that domains entered in a `cdns-rate-limit add` subcommand must refer to a zone delegation point.

### Examples

```
nrcmd> cdns-rate-limit enable domain-rate-limiting
nrcmd> cdns-rate-limit set client-rate-limiting-qps=1000
nrcmd> cdns-rate-limit add example.com both 500
nrcmd> cdns-rate-limit modify 0 rate-limit=750
```

### Attributes

`client-rate-limiting` [bool](#) default = off

Enables ip based client rate limiting.

`client-rate-limiting-factor` [rangeint](#)(0-100) default = 10

When `client-rate-limiting` is enabled and a client is being rate limited, specifies that one out of this number of queries from that client will be allowed to complete.

`client-rate-limiting-qps` [rangeint](#)(1-2147483647) default = 1000

Specifies the rate limit for incoming DNS clients.

`client-report-max-count` [rangeint](#)(10-1000) default = 10

Specifies the maximum number of entries in the list of rate limited clients. This limit is applied to the lists of clients that are logged, returned as part of activity summary or included in statistics.

`description` [string](#)

A description of the DNS rate limit configuration.

**domain-rate-limiting** [bool](#) default = off

Enables rate limiting for name server zones.

**domain-rate-limiting-factor** [rangeint](#)(0-100) default = 10

When domain-rate-limiting is enabled and a zone is being rate limited, specifies that one out of this number of queries to the specified zone will be allowed to complete.

**domain-rate-limiting-list** [nlist](#)(obj(0))

Specifies a list of domains that use a rate limit other than domain-rate-limiting-qps. The list entries have the following attributes;

- domain - The name of the zone delegation point to which this entry applies.
- applies-to - Specifies if this entry applies to only the zone designated by 'domain', only zones specified by subdomains of 'domain' or both.
- rate-limit - The rate limit that applies to zones covered by this entry.

**domain-rate-limiting-qps** [rangeint](#)(1-2147483647) default = 1000

Specifies the rate limit for name server zones.

**domain-report-max-count** [rangeint](#)(10-1000) default = 10

Specifies the maximum number of entries in the list of rate limited domains. This limit is applied to the lists of domains that are logged, returned as part of activity summary or included in statistics.

## cdns-forwarder

cdns-forwarder - Controls and configures DNS Forwarders in the DNS Caching server

### Synopsis

```
cdns-forwarder <name> create <attribute>=<value>
cdns-forwarder <name> delete
cdns-forwarder <name> show
cdns-forwarder <name> get <attribute>
cdns-forwarder <name> set <attribute>=<value>
cdns-forwarder <name> unset <attribute>
cdns-forwarder list
cdns-forwarder listnames
cdns-forwarder listbrief
```

### Description

The cdns-forwarder command lets you create and edit DNS Forwarder objects. Typically forwarders are other DNS Caching servers that have access to internet or external DNS resources. The DNS Caching server must be reloaded for changes to take effect.

The <name> specifies the domain that this forwarder will apply to (i.e. example.com). In order to forward recursive queries for all domains, create a forwarder object with <name> ".".

The forwarder <addr> or <host> must also be specified in order for the forwarders to be properly configured. Network Registrar recommends using the <addr> which can be set to one more (separated by commas) ipv4 or ipv6 addresses followed by an optional port number (i.e. <addr>[@<port>]).

### Examples

```
nrcmd> cdns-forwarder example.com create
nrcmd> cdns-forwarder example.com set addr=10.10.10.1
nrcmd> cdns-forwarder example.com set addr=10.10.10.1,10.10.10.2
nrcmd> cdns-forwarder example.com show
nrcmd> cdns-forwarder example.com delete
```

### Status

## See Also

[cdns-exception](#)

### Attributes

**addr** [nlist](#)(obj(0))

Specifies the IP addresses and ports of the name servers.  
Can be IP v4 or IP v6.

**cisco-umbrella** [bool](#) default = false

Specifies that this forwarder definition is for Cisco Umbrella DNS servers.  
Cisco Umbrella IP addresses:  
208.67.222.222 + 208.67.220.220  
2620:119:35::35 + 2620:119:53::53

**description** [string](#)

An optional description for this forwarder configuration.

**host** [nlist](#)(obj(0))

Specifies the host names of the name servers for this set. They are resolved before they are used.

**name** [dname](#) required,unique

Specifies which domain of queries that should be forwarded for to this DNS server for resolution. To forward queries for all domains, specify the root domain '.'.

**tls** [bool](#) default = disabled

Enables TLS transport for this forwarder. If you enable this, also configure a `tls-cert-bundle` to load CA certs, otherwise the connections cannot be authenticated. Default is disabled.

**tls-auth-name** [string](#)

Defines the auth name for the forwarder server. If TLS is enabled, the `cdns` server will check the TLS authentication certificates with that name sent by the forwarder server.

## cdns-exception

`cdns-exception` - Controls and configures DNS Exceptions in the DNS Caching server

### Synopsis

```
cdns-exception <name> create <attribute>=<value>  
cdns-exception <name> delete  
cdns-exception <name> show  
cdns-exception <name> get <attribute>  
cdns-exception <name> set <attribute>=<value>  
cdns-exception <name> unset <attribute>  
cdns-exception list  
cdns-exception listnames  
cdns-exception listbrief
```

### Description

The `cdns-exception` command lets you create and edit DNS Exception objects. Exceptions refer to other DNS name servers that are authoritative for the exception domain(s). Typically exceptions are required when accessing internal DNS name servers.

The `<name>` specifies the domain that this exception will apply to (i.e. `example.com`).

The exception `<addr>` or `<host>` must also be specified in order for the exceptions to be properly configured. Network Registrar recommends using the `<addr>` which can be set to one more (separated by commas) `ipv4` or `ipv6` addresses followed by an optional port number (i.e. `<addr>[@<port>]`).

The DNS Caching server must be reloaded for changes to take effect.

## Examples

```
nrcmd> cdns-exception example.com create
nrcmd> cdns-exception example.com set addr=10.10.10.1
nrcmd> cdns-exception example.com set addr=10.10.10.1,10.10.10.2
nrcmd> cdns-exception example.com show
nrcmd> cdns-exception example.com delete
```

## Status

## See Also

[cdns-forwarder](#)

## Attributes

addr [nlist](#)(obj(0))

Specifies the IP addresses and ports of the name servers.  
Can be IP v4 or IP v6.

description [string](#)

An optional description for this exception.

host [nlist](#)(obj(0))

Specifies the host names of the name servers for this set. They are resolved before they are used.

name [dname](#) required,unique

Specifies the domain of queries that should be sent to this exception DNS server for resolution.

prime [bool](#) default = off

Specifies if the DNS caching server performs NS set priming, which is similar to root hints, where it starts using the list of name servers currently published by the zone. Thus, if the hint list is slightly outdated, the resolver picks up a correct list online.

tls [bool](#) default = disabled

Enables TLS transport for this exception. If you enable this, also configure a `tls-cert-bundle` to load CA certs, otherwise the connections cannot be authenticated. Default is disabled.

tls-auth-name [string](#)

Defines the auth name for the exception server. If TLS is enabled, the cdns server will check the TLS authentication certificates with that name sent by the exception server.

uses-views [bool](#) default = false

Indicates whether or not the exception server uses zone views.

## certificate

certificate - Controls and configures SSL/TLS Certificates

## Synopsis

```
certificate <name> create <type> file=<certificate-file>
                                     [<attribute>=<value>...]
certificate <name> delete
certificate <name> show
certificate <name> get <attribute>
certificate <name> set <attribute>=<value>
certificate <name> unset <attribute>
certificate list
certificate listnames
certificate listbrief

certificate < <name> | all > pull < ensure | replace | exact >
                                     <cluster-name> [-report-only | -report]
```

```
certificate < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
certificate <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The certificate command lets you create and manage SSL/TLS Certificate objects. Certificate objects can be used in conjunction with DNS over TLS (DoT) or securing a CNR Web UI connection. Certificate objects are monitored for expiration.

The <type> specifies the component the certificate is will apply to. Valid values are webui, adns or cdns.

The <certificate-file> is the path to an SSL/TLS certificate file. This file should be unencrypted if it is to be used with the Authoritative or Caching DNS servers.

Most of the attributes on the certificate object are automatically populated from the certificate contents and cannot be set or modified.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

```
nrcmd> certificate example create webui file=cert.pem
nrcmd> certificate example set description=example
nrcmd> certificate example get subject
nrcmd> certificate example show
nrcmd> certificate example delete
```

## Status

## See Also

### Attributes

**certificate-contents** [string](#) immutable

Specifies the base64 encoded certificate contents read from the certificate file.

**description** [string](#)

A description of the certificate being managed.

**issuer** [string](#) immutable

Specifies information about the entity that issued the certificate.

**name** [string](#) required,unique

The name of the certificate being managed.

**public-key-algorithm** [string](#) immutable

Specifies the algorithm and size of the public key.

**serial-number** [blob](#) immutable

Specifies the serial number of the certificate.

**signature-algorithm** [string](#) immutable

Specifies the algorithm and size of the signature.

**subject** [string](#) immutable

Specifies information about the entity receiving the certificate.

**type** `enumint`(none=0, webui=1, adns=2, cdns=3) required,immutable

Specifies the Cisco Network Registrar component that that will use the certificate.

**validity-not-after** `date` immutable

Specifies the date and time marking the end of the certificate validity period.

**validity-not-before** `date` immutable

Specifies the date and time marking the start of the certificate validity period.

**version** `int` immutable

Specifies the SSL version for the certificate.

## client

`client` - Creates clients and assigns them to client-classes

### Synopsis

```
client <name> create [<attribute>=<value>]
client <name> delete
client list
client listnames
client listbrief
client <name> show
client <name> get <attribute>
client <name> set <attribute>=<value>
client <name> unset <attribute>
```

### Description

The `client` command assigns attributes to a specific client entry. These attributes determine what type of IP address, policy, or both that Network Registrar assigns to the requesting host. Network Registrar always stores the client identifier (MAC address or default) in lowercase characters.

Because the DHCP server reads the client-specific configuration information each time a request comes in, you do not have to reload the server after modifying it. However, you must reload the server if you modify the default client configuration.

The attributes you can assign include such things as a class of client, a policy, an action, and the inclusion or exclusion of scope selection tags. The DHCP server looks up these properties to determine how it should process a host request for an IP address.

If you have common client attributes to configure, such as selection criteria, use the `client-class` so that multiple client configurations can reference the attributes.

You can specify the client by using the MAC address or some other unique client identifier related to the `client-lookup-id` that is specified in the client's associated `client-class`.

A sample Ethernet MAC address might be `1,6,00:a0:24:2e:9c:20`

```
client name create [attribute=value...]
```

```
client default create [attribute=value...]
```

Creates the client identifier as a MAC address or the word `default` (and optionally defines its attributes). The default client configuration applies to all clients that do not have an explicit configuration. If an entry for the client already exists, the command overwrites it.

If using a MAC address, it should be in the form `hardware, length, address` (without spaces and including the commas):

`hardware`

Usually 1 (Ethernet) or 6 (Token Ring), but can be any number from 1 through 255.

length  
Octets in the MAC address (usually 6, but can be any number from 1 through 16).

address  
MAC address itself, with octets separated by colons, and each octet having a two-character hex value from 00 through FF (not case-sensitive).

## Examples

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b create client-class-name=external
```

## Status

## See Also

[client-class](#)

## Attributes

**action flags**(exclude=1, deprecated-one-shot=2, deprecated-use-release-grace-period=3, none=32)

Describes the action the DHCP server takes for this client.

- 1 exclude - causes the server to ignore all communication from this client.
- 2 deprecated-one-shot - now deprecated.
- 3 deprecated-use-release-grace-period - now deprecated.
- 32 none

If you specify the exclude action in the default client entry, then any client not specifically registered through the client command cannot communicate with the server. Note: The deprecated flags (2,3) are now available through the policy command attributes inhibit-all-renews and release-grace-period.

**add-to-environment-dictionary** [string](#)

Lists attribute-value pairs that are added to the environment dictionary whenever this client-class is associated with an incoming DHCP request. You can use these attribute-value pairs to configure extensions or expressions without having to rewrite the executable code in the extension or expression. The string must have the format: "attribute1=value1,attribute2=value2, ... ,attributen=valuen"

**authenticate-until** [date](#)

Sets an authentication expiration date, using date format or the forever keyword. Dates can be in the 2h (two hours ago, for example) or month day hour:minute[:second] year format. Formats for the date are:

- + Time in the future, where num is a decimal number and unit is s, m, h, d, or w for seconds, minutes, hours, days or weeks, respectively.
- :[:] Month, day, 24-hour time, and 2-or-4-digit-year. For example: Jun 30 20:00:00 2007. Enter the time that is local to the nrcmd process.
- forever Does not expire the authentication for this client.

**client-class-name** [nameref](#)(0)

Identifies the client-class to which a client belongs. If the client is not a member of a client-class, then the DHCP server uses the default client-class properties.

**default-vpn** [nameref](#)(0)

Names the VPN to assign to clients that do not already have a vpn-id or vrf-name value.

**domain-name** [string](#)

Gives the domain name (which must be a zone) to use when performing DNS updates. Places the client's A record in this DNS domain. This feature is maintained for compatibility with prior versions. Additional options to specify the forward zone are provided on the client policy (or embedded policy) and its referenced DNSUpdateConfig objects.

**embedded-policy** [obj](#)(0)

Specifies the embedded policy object for this client.

**host-name** [string](#)



Specifies the hostname. Use this string to replace any hostname DHCP option that the DHCP client sends. The two forms for specifying the hostname are:

1. A string that does not start with an at (@) sign. This form of host-name value is used to override the DHCP client request host name. When you enter a valid name, you cause the DHCP server to ignore any host-name specified by this client, and, instead, use this client-entry attribute. The actual value of the hostname option in the client's DHCP packet is ignored. You can use any valid DNS name. You cannot use underscores ( ) in the hostname.
2. A string that starts with an at (@) sign. Network Registrar uses this form of hostname value to signal the following special handling:
  - @host-name-option  
Causes the server to use whatever hostname option the client sent. This is the default behavior if there is no entry for hostname in either the client or client-class.
  - @no-host-name-option  
Causes the server to drop the hostname option that the client sent, and not replace it. If you have disabled DNS name synthesis, then the client will have no name placed into DNS.
  - @use-macaddress  
Causes the server to synthesize a hostname for the client that is derived from its MAC address, and is thus unique. This token is used to ensure that a client has a valid name in DNS.

This feature is maintained for compatibility with earlier versions. Additional options for hostname synthesis are provided on the DNSUpdateConfig object referenced by the policy hierarchy.

#### **over-limit-client-class-name** [string](#)

Identifies which client-class to use if this client is over the limit allowed for the number of simultaneous active leases with a common limitation-id.

#### **override-vpn** [nameref\(0\)](#)

Names the VPN to assign to clients, no matter what values clients present as VPN-IDs or vrf-names.

#### **policy-name** [nameref\(0\)](#)

Identifies the policy to add to Network Registrar's DHCP policy search list for this client.

#### **reserved-addresses** [nlist\(obj\(0\)\)](#)

Specifies the list of addresses reserved for the client. The first available address to match a usable Scope (which must have restrict-to-reservations enabled) will be assigned to the client.

#### **reserved-ip6addresses** [nlist\(obj\(0\)\)](#)

Specifies the list of addresses reserved for the client. All available addresses to match a usable Prefix (which must have restrict-to-reservations enabled) will be assigned to the client.

#### **reserved-prefixes** [nlist\(obj\(0\)\)](#)

Specifies the list of prefixes reserved for the client. All available prefixes to match a usable Prefix (which must have restrict-to-reservations enabled) will be assigned to the client.

#### **selection-criteria** [nlist\(obj\(0\)\)](#)

Lists selection tags for this client. All the criteria in this list must appear in the scope/prefix selection tags for a scope/prefix to be considered acceptable to this client.

#### **tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### **unauthenticated-client-class-name** [string](#)

Identifies the client-class to use if this client is no longer authenticated.

#### **user-defined** [string](#)

Contains an opaque user-defined string that can be set and queried. This attribute has no effect on the operation of the DHCP server.

#### **userid** [string](#)

Contains the username owning the device. This attribute has no effect on the operation of the DHCP server.

## **client-class**

client-class - Creates client-classes

## Synopsis

```
client-class <name> create [<attribute>=<value>]
client-class <name> delete
client-class list
client-class listnames
client-class listbrief
client-class <name> show
client-class <name> set <attribute>=<value> [<attribute>=<value> ...]
client-class <name> get <attribute>
client-class <name> unset <attribute>

client-class < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
client-class < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
client-class <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The `client-class` command applies a set of attributes to a group or class of DHCP client configurations. Unlike most client configurations, the DHCP server reads the `client-class` configurations at server startup time. Therefore, you must reload the server for changes to take effect.

You must enable `client-class` processing for the server for Network Registrar to recognize client-classes, as the examples show.

The `pull`, `push` and `reclaim` commands are only available when connected to a regional cluster. `push` and `reclaim` allow a list of clusters or "all".

## Examples

```
nrcmd> dhcp enable client-class
nrcmd> client-class internal create
nrcmd> dhcp reload
```

## Status

## See Also

### Attributes

**action** [flags](#)(exclude=1, deprecated-one-shot=2, deprecated-use-release-grace-period=3, none=32)

Specifies what action to take with this `client-class`. You can specify `exclude`, causing the server to ignore all communication from this client.

If you specify the `exclude` action in the default client entry, then any client not specifically registered through the `client` command cannot communicate with the server.

The `one-shot` and `use-release-grace-period` actions are now deprecated and ignored; these are available through the policy `inhibit-all-renews` and `release-grace-period` attributes.

**add-to-environment-dictionary** [string](#)

This string contains attribute-value pairs that are added to the environment dictionary whenever this `client-class` is associated with an incoming DHCP request. You can use these attribute-value pairs to configure extensions or expressions without having to re-write the executable code in the extension or expression. The string must have the format:  
"attribute1=value1,attribute2=value2, ... ,attributen=valuen"

**client-lookup-id** [expr](#)

Specifies the key value used to lookup the specified client in the client database, using an expression that evaluates to a string or a blob that is a valid string. The lookup can be local or through LDAP.

**default-vpn** [nameref](#)(0)

Determines the VPN to which a client is assigned if the client does not supply a VPN-ID (or `vrf-name`) value.

**domain-name** [string](#)

Sets the domain name, which must be a zone, for performing DNS updates. Places the client's A record in this DNS domain. This feature is maintained for compatibility with prior versions. Additional options to specify the forward zone are provided on the client-class's policy (or embedded policy) and its referenced DNSUpdateConfig objects.

**embedded-policy** [obj\(0\)](#)

Specifies the embedded policy object for this client-class.

**host-name** [string](#)

Specifies the hostname for clients in this client-class. Network Registrar uses this form of hostname value to signal the following special handling:

**@host-name-option**  
Causes the server to use whatever hostname option the client sent. This is the default behavior if there is no entry for host-name in either the client or client-class.

**@no-host-name-option**  
Causes the server to drop the hostname option that the client sent, and not replace it. If you have disabled DNS name synthesis, then the client will have no name placed into DNS.

**@use-macaddress**  
Causes the server to synthesize a host-name for the client that is derived from its MAC address, and is thus unique. This token is used to ensure that a client has a valid name in DNS.

This feature is maintained for compatibility with prior versions. Additional options for host-name synthesis are provided on the DNSUpdateConfig object referenced by the policy hierarchy.

**limitation-id** [expr](#)

Specifies an expression that evaluates to a binary large object (blob), or a string that can be used as a blob. The resulting value groups leases that have a maximum limit on the number of simultaneous active leases allowed. To configure the limit, use limitation-count attribute of the policy command. See also the over-limit-client-class attribute.

**over-limit-client-class-name** [string](#)

Designates the client-class used if this client is over the limit allowed for the specified limitation-id.

**override-client-id** [expr](#)

Specifies the client-identity value for the specified client, using an expression that evaluates to a binary large object (blob). The value that is derived from the expression evaluation replaces any client-id option value in the incoming packet (though in actual practice, both values are retained in the lease-state database).

**override-vpn** [nameref\(0\)](#)

Determines the VPN to which a client is assigned regardless of what the client provides for a VPN-ID (or vrf-name) value.

**policy-name** [nameref\(0\)](#)

Names the policy that should be used for this client-class.

**selection-criteria** [nlist\(obj\(0\)\)](#)

Lists the selection tags for this client-class. All the criteria in this list must appear in the scope/prefix selection tags for a scope/prefix to be considered acceptable to this client-class.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**unauthenticated-client-class-name** [string](#)

Names the client-class used if this client is no longer authenticated.

**user-defined** [string](#)

Provides an opaque user-defined string that can be set and queried. This property has no effect on the operation of the DHCP server.

**v6-client-lookup-id** [expr](#)

Specifies the key value used to lookup the DHCPv6 client in the client database, using an expression that evaluates to a string or a blob that is a valid string. The lookup can be local or through LDAP.

**v6-override-client-id** [expr](#)

An expression which evaluates to a blob which is used for the client-identity value for the current DHCPv6 client. Conceptually this value derived from the expression evaluation will replace any

client-id option value in the incoming packet (though in actual practice, both values are retained in the lease-state database).

## client-class-policy

client-class-policy - Adds DHCP policy information to a client-class

### Synopsis

```
client-class-policy <name> delete
client-class-policy <name> set <attribute>=<value>
    [<attribute>=<value> ...]

client-class-policy <name> get <attribute>

client-class-policy <name> disable <attribute>
client-class-policy <name> enable <attribute>
client-class-policy <name> show

client-class-policy <name> setLeaseTime <time-val>
client-class-policy <name> getLeaseTime

client-class-policy <name> setOption <opt-name | id> <value>
    [-blob | -expression] [-roundrobin]
client-class-policy <name> getOption <opt-name | id>
client-class-policy <name> unsetOption <opt-name | id>
client-class-policy <name> listOptions

client-class-policy <name> setV6Option <opt-name | id>[.<instance>]
    <value> [-blob | -expression] [-roundrobin]
client-class-policy <name> addV6Option <opt-name | id>[.<instance>]
    <value> [-blob | -expression] [-roundrobin]
client-class-policy <name> getV6Option <opt-name | id>[.<instance>]
client-class-policy <name> unsetV6Option <opt-name | id>[.<instance>]
client-class-policy <name> listV6Options

client-class-policy <name> setVendorOption <opt-name | id>
    <opt-set-name> <value> [-blob | -expression]
client-class-policy <name> getVendorOption <opt-name | id>
    <opt-set-name>
client-class-policy <name> unsetVendorOption <opt-name | id>
    <opt-set-name>
client-class-policy <name> listVendorOptions

client-class-policy <name> setV6VendorOption <opt-name | id>
    <opt-set-name> <value> [-blob | -expression]
client-class-policy <name> getV6VendorOption <opt-name | id>
    <opt-set-name>
client-class-policy <name> unsetV6VendorOption <opt-name | id>
    <opt-set-name>
client-class-policy <name> listV6VendorOptions
```

### Description

The client-class-policy command configures embedded policies for client-classes. Each client-class can contain option data in its embedded policy and can refer to a named policy with more option data; for example, a router IP address.

An embedded policy is a collection of DHCP option values and settings associated with (and named by) a client-class. Network Registrar implicitly creates and deletes an embedded client-class policy when you create and delete the corresponding client-class. You manipulate the client-class policy using the name of the client-class to which the embedded policy is attached.

See the help file for the policy command for more information.

### Examples

### Status

## See Also

[policy](#), [client-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the `inhibit-all-renews` attribute or the `inhibit-renews-at-reboot` attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

### allow-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

### default-prefix-length [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

### excluded-prefix [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603). Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified. Using the example in RFC 6603, this attribute would be set to `0:0:0:f::/64`. If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

### forward-dnsupdate [nameref](#)(0)

Specifies the name of the update configuration that determines

which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1).  
The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server.  
The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server.  
The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only.

Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- |   |          |   |
|---|----------|---|
| 1 | allow    | Allows clients to request reconfiguration support and the server will honor the request (default).  |
| 2 | disallow | Allows clients to request reconfiguration support but the server will not honor the clients' request.   |
| 3 | require  | Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option. |

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

**reconfigure-via-relay** [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

**reverse-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### **server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### **shortest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### **split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### **tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### **unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system\_default\_policy, then the default is 86400 seconds (or 24 hours).

#### **use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### **v4-bootp-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all BOOTP clients.

#### **v4-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### **v6-max-client-preferred-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### **v6-max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### **v6-max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### **v6-max-client-valid-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it



does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsoo-allowed-options [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## client-policy

client-policy - Adds DHCP policy information to a client object

### Synopsis

```
client-policy <name> delete
client-policy <name> set <attribute>=<value>
                        [<attribute>=<value> ...]
client-policy <name> get <attribute>

client-policy <name> disable <attribute>
client-policy <name> enable <attribute>
client-policy <name> show

client-policy <name> setLeaseTime <time-val>
client-policy <name> getLeaseTime

client-policy <name> setOption <opt-name | id> <value>
                        [-blob | -expression] [-roundrobin]
client-policy <name> getOption <opt-name | id>
client-policy <name> unsetOption <opt-name | id>
client-policy <name> listOptions

client-policy <name> setV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
client-policy <name> addV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
client-policy <name> getV6Option <opt-name | id>[.<instance>]
client-policy <name> unsetV6Option <opt-name | id>[.<instance>]
client-policy <name> listV6Options

client-policy <name> setVendorOption <opt-name | id>
                        <opt-set-name> <value> [-blob | -expression]
client-policy <name> getVendorOption <opt-name | id>
                        <opt-set-name>
client-policy <name> unsetVendorOption <opt-name | id>
                        <opt-set-name>
client-policy <name> listVendorOptions

client-policy <name> setV6VendorOption <opt-name | id>
                        <opt-set-name> <value> [-blob | -expression]
client-policy <name> getV6VendorOption <opt-name | id>
                        <opt-set-name>
client-policy <name> unsetV6VendorOption <opt-name | id>
                        <opt-set-name>
```

```
client-policy <name> listV6VendorOptions
                    <value> [-blob | -expression] [-roundrobin]
```

## Description

The `client-policy` command configures embedded policies for clients. Each client can contain option data in its embedded policy and might refer to a named policy with more option data - for example, a router IP address. Network Registrar implicitly creates and deletes an embedded client policy when you create or delete the corresponding client. You manipulate the client policy using the name of the client to which the embedded policy is attached.

See the help file for the `policy` command for more information.

## Examples

## Status

## See Also

[policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the `inhibit-all-renews` attribute or the `inhibit-renews-at-reboot` attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are

checked.  
The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses.  
The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it.  
The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603).  
Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified.  
Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64.  
If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebind operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used.  
The default is the value of the default-prefix-length.  
This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this

value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, It overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

1 allow	Allows clients to request reconfiguration support and the server will honor the request (default).
2 disallow	Allows clients to request reconfiguration support but the server will not honor the clients' request.
3 require	Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not

include a Reconfigure-Accept option. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

#### reconfigure-via-relay [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

#### reverse-dnsupdate [nameref\(0\)](#)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### server-lease-time [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### shortest-prefix-length [rangeint\(0-128\)](#)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### split-lease-times [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### unavailable-timeout [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system default\_policy, then the default is 86400 seconds (or 24 hours).

#### use-client-id-for-reservations [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### v4-bootp-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

#### v4-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### v6-max-client-preferred-lifetime [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value.

This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### v6-max-client-rebinding-time [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### v6-max-client-renewal-time [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### v6-max-client-valid-lifetime [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsso-allowed-options [nlist](#)(obj(0))

Lists the RS00-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RS00 (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## cluster

cluster - Configures the local and remote clusters

### Synopsis

```
cluster <name> create <address | ipv6-address> [<attribute>=<value> ...]
cluster <name> delete
cluster list
cluster listnames
cluster listbrief
cluster <name> show
cluster <name> set <attribute>=<value> [<attribute>=<value> ...]
cluster <name> get <attribute>
cluster <name> unset <attribute>

cluster <name> enable <attribute>
cluster <name> disable <attribute>
```

```

cluster <name> activate
cluster <name> deactivate
cluster <name> resynchronize
cluster <name> sync
cluster <name> updateReplicaData
cluster <name> removeReplicaData
cluster <name> recoverData
cluster <name> pollLeaseHistory
cluster <name> getLeaseHistoryState
cluster <name> pollSubnetUtilization
cluster <name> viewReplicaData < <class-name> | <cli-command> >
                        [-listbrief | -listcsv]

```

## Description

The cluster command configures the specified local or remote cluster, primarily providing connection and polling information; for example, IP address, fully qualified domain name, and HTTP port. This information provides reference points for objects related to the cluster.

Activate, deactivate, resynchronize, sync, updateReplicaData, removeReplicaData, recoverData, pollLeaseHistory, getLeaseHistoryState, pollSubnetUtilization, and viewReplicaData are only available when connected to a regional cluster.

## Examples

## Status

## See Also

### Attributes

**addrutil-poll-interval** [rangetime](#)(0-1y)

Specifies how often to collect subnet utilization data from the DHCP server for this cluster. If set to 0, polling does not occur.

**addrutil-poll-offset** [rangetime](#)(0-24h)

Provides a fixed time of day for subnet utilization polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided the polling interval is less than 24 hours, and the offset value is less than the polling interval. If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for polling will ensure that the first polling event occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

**addrutil-poll-retry** [rangeint](#)(0-4)

Controls how often to retry if subnet utilization polling fails.

**admin** [string](#)

Sets the administrator identity to use to contact this cluster.

**cluster-id** [short](#) unique,immutable

Identifies the local cluster that is the authoritative source for this object. This attribute is set by the CCM server.

**fqdn** [dname](#)

Provides the fully qualified domain name of this server. This attribute is not used to contact the cluster.

**http-port** [int](#)

Specifies the http-port to use for non-SSL-secured connections to the web server for this cluster.

**https-port** [int](#)

Controls the https-port to use for SSL-secured connections to the webserver for this cluster. This port is only used if the value of the use-https-port attribute is true.

**ip6address** [ip6addr](#)

Provides the IPv6 address of this server. This attribute, rather than the fqdn is used to connect to the cluster.

**ipaddr** [ipaddr](#)

Provides the IP address of this server. This attribute, rather than the fqdn is used to connect to the cluster.

**lease-hist-poll-interval** [rangetime](#)(0-1y)

Specifies how often to collect the lease history from the DHCP server for this cluster. If set to 0, polling does not occur.

**lease-hist-poll-offset** [rangetime](#)(0-24h)

Provides a fixed time of day for lease history polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided the polling interval is less than 24 hours, and the offset value is less than the polling interval. If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for polling will ensure that the first polling event occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

**lease-hist-poll-retry** [rangeint](#)(0-4)

Controls how often to retry if lease history polling fails.

**licensed-services** [flags](#)(dhcp=17, dns=18, cdns=19)

Identifies the component services that are licensed for this local cluster.

**local-servers** [nlist](#)(obj(0)) transient

Lists the servers associated with this cluster. This transient attribute makes it easier for clients that want to show a tree of clusters with their child servers to get all the information in a single request.

**name** [string](#) required,unique

Names the cluster.

**password** [clrtxt](#)

Sets the password that authenticates the identity stored in the admin attribute. Do not use this clear-text value except within process memory. Use the corresponding password-secret instead.

**password-secret** [secret](#)

Sets the identifier of the secret representing the password that authenticates the identity stored in the admin attribute.

**poll-replica-rrs** [enumint](#)(none=0, all=1, protected=2)

Specifies the RRs to replicate for primary zones. If set to none, no RR data will be replicated for this cluster. If unset, the CCM server setting will apply.

**product-version** [pcv](#)

Displays the cluster product version number in major, minor, rev form. This value is updated when the cluster is resynchronized.

**remote-id** [short](#)

Sets the ID on the remote cluster that refers back to the local cluster. If there are two cluster objects on two servers that share a secret and refer to each other, then the local ID = the remote-id, the local remote ID = the remote ID, and the value of the local shared secret = the value the remote shared secret.

**replica-poll-interval** [time](#) default = 4h

Set the automatic replication interval; that is, how often Network Registrar polls this server for replica data.

**replica-poll-offset** [time](#) default = 4h

Sets a fixed time of day for replica polling. This time is interpreted as a time of day offset, with 0 being 12 midnight, provided that:  
the polling interval is less than 24 hours, and  
the offset value is less than the polling interval.  
If the offset value is greater than the polling interval, or the interval is greater than 24 hours, the offset is ignored.



The scheduler for polling ensures that the first polling event occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm.

**replication-initialized** [bool](#) default = false

Indicates whether data replication has already been initialized on this cluster.

**restore-state** [enumshort](#) default = active, transient

Indicates whether the cluster has been deactivated or is in the process of being restored from the replica db.

**scp-port** [int](#)

Controls the port number used for SCP communications.

**scp-read-timeout** [time](#) default = 20m

Specifies the time limit to wait for a response when reading an SCP message from this cluster.

**shared-secret** [secret](#)

Specifies the identifier for the secret shared between the server storing this object and the cluster it represents. This shared secret is used to generate single-sign-on authentication tokens.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**use-https-port** [bool](#) default = false

Controls whether the https-port is used to make single sign-on connections to the cluster. If the value is false, or the https-port attribute is not set, then the http-port is used.

**use-ssl** [enumstr](#)(disabled=1, optional=2, required=3) default = optional

Specifies the security mode should we use when connecting to this cluster. If optional, and the security library is installed, we will try to secure the connection. If required, we will not make the connection unless we can secure the connection (this requires the security library to be installed). If none, we will not try to secure the connection.

## debug

debug - Sets debug level

### Synopsis

```
[no] debug smart_lic <error|trace|debug|all>
```

### Description

The **debug smart\_lic** command allows you to set the smart license debug level. When executed with 'no' as prefix, unsets the specific debug level.

This command is available in smart license configuration mode.

### Examples

### Status

### See Also

[smart](#)

## dhcp

dhcp - Configures and controls the DHCP server

## Synopsis

```
dhcp disable <attribute>
dhcp enable <attribute>

dhcp get <attribute>
dhcp set <attribute>=<value> [<attribute>=<value> ...]
dhcp unset <attribute>
dhcp show

dhcp getStats [ <all | server [,] failover [,] dhcpv6 [,] top-utilized>
               [total | sample] ]
dhcp resetStats
dhcp getScopeCount [vpn <name> | all]
dhcp getPrefixCount [vpn <name> | all]

dhcp attachExtension <extension-point> <extension-name> [<sequence-number>]
dhcp detachExtension <extension-point> [<sequence-number>]
dhcp listExtensions

dhcp getRelatedServers [column-separator=<string> | full]
dhcp getRelayState [all] [full]

dhcp getRenewalData [<max-buckets>]

dhcp updateSms [all]
dhcp serverLogs show
dhcp serverLogs nlogs=<nlogs> logsize=<logsize>

dhcp limitationList <ipaddr> [<limitation-id>] show
```

## Description

The dhcp command lets you configure the DHCP server in a cluster.

```
dhcp getStats [<statistics-categories> [total | sample]]
dhcp resetStats
```

The getStats command retrieves statistics from a running DHCP server. You can supply one or more specific categories of statistics counters (server, failover, dhcpv6, top-utilized), or the keyword all to retrieve all supported categories. If collection of sample counters is enabled in the server, you can retrieve the most recent sample counters instead of the running totals by specifying sample after the categories. The resetStats command resets the running totals counters.

```
dhcp attachExtension <extension-point> <extension-name> <sequence-number>
dhcp detachExtension <extension-point> <sequence-number>
dhcp listExtensions
```

Use the commands attachExtension, detachExtension, and listExtensions to configure the extensions points in the server.

You can associate multiple extensions with each extension point, and each executes in the order specified by the sequence number used when the attachment was made. If no sequence number is used with attachExtension and detachExtension, it defaults to 1. If multiple extensions are configured for a given point, listExtensions shows the sequence numbers associated with each. Sequence numbers must be in the range 1-32.

The available extension points are:

The attachExtension command sets the specified extension point (and optional sequence position) to call the named extension. If the extension point is already configured (for a given sequence position) to call an extension, Network Registrar overwrites it with the new value.

The detachExtension command removes any extension configuration from the specified extension point and sequence number.

The listExtensions command shows the current configurations for each extension point.

You can put the DHCP server into import mode by enabling the import-mode feature and then restarting the server. You take the server out of import-mode by disabling the feature and restarting the server. You can use import mode to exclude all DHCP lease requests except for the specially tagged ones that come from the

CLI during lease import (see the import command).

**dhcp getRelatedServers [column-separator=<string>]**

The dhcp getRelatedServers command displays a table with the following information for each associated failover, DNS or LDAP server:

Type  
Main, Backup, DNS or LDAP

Name  
DNS host name

Address  
IP Address in dotted octet format

Communications  
OK or INTERRUPTED

Requests  
Number of outstanding requests

<cluster-name> State  
Failover state of this server

Partner State  
Failover state of partner

**dhcp getRelatedServers full**

The full list of related servers objects is displayed (rather than a table).

**dhcp getRelayState [all] [full]**

Only useful if failover is being used and the health-checking feature has been enabled. Reports on the state of communications between the failover partner and each relay agent. If **all** is not specified, only the relays that appear to be having communications issues with the failover partner are reported (i.e., those in the interrupted state). If **full** is specified, the objects are displayed (rather than a table).

**dhcp getRenewalData [<max-buckets>]**

Reports information related to the distributed renewals feature. By default, the expected number of client renewals over time is shown in at most 20 buckets but this can be overridden by specifying the number desired.

**dhcp updateSms [all]**

The dhcp updateSms command initiates SMS processing. To send all leases to SMS, use the argument all; otherwise, only the new leases activated since the last time the command ran successfully are sent. To run this command, turn on sms-network-discovery and set sms-library-path. The command returns an error if sms-network-discovery is not turned on or if it is unable to load SMS library or if the optional argument string is invalid, otherwise it returns success to indicate SMS processing started successfully.

**dhcp serverLogs show**

The serverLogs show command displays the number of log files and the maximum size for each file.

The serverLogs command allows setting the two server logging parameters, nlogs and logsize. Either or both may be specified in the command, and changes will only occur to the one(s) specified. When setting logsize, the value may be suffixed with K or M to signify units of thousands or millions. Note that in order for these changes to take effect you must save the changes and restart the server Agent.

```
dhcp serverLogs nlogs=6 logsize=500K
dhcp serverLogs logsize=5M
```

**dhcp getScopeCount [vpn <name> | all]**

**dhcp getPrefixCount [vpn <name> | all]**

The getScopeCount command displays the scopes, networks, and VPNs for the current VPN, all VPNs, or a specific VPN.

The getPrefixCount command displays the prefixes, links, and VPNs for the current VPN, all VPNs, or a specific VPN.

**dhcp limitationList <ipaddr> [<limitation-id>] show**

Lists DHCP clients and leases that are associated by a common limitation-id for the client (see the client command). Use this command when a DHCP client is denied service because the number of existing clients with a common limitation-id equals the allowed

limitation-count, as set for a policy (see the policy command). It then determines which existing clients with that limitation-id have active leases.

If you specify both the ipaddr and limitation-id arguments, the ipaddr determines the network in which to search, and does not have to be an actual IP address that the DHCP server could allocate. In this case, the limitation-id must be a blob in nn:nn:nn format (such as 01:02:03) or a string in string format. If you omit the limitation-id, the ipaddr must be the IP address of a currently active lease, and the limitation-id used for the command will be the one associated with that lease.

If you want to determine the existing clients and leases using up the limitation-count for a particular limitation-id because the following message appeared in the DHCP server log:

```
Warning Server 0 05646 Could not add Client MAC:
'1,6,01:02:03:04:0c:03' with limitation-id: 01:02:03
using Lease: 10.0.0.23, already 3 Clients with that id.
No over-limit client class specified! Dropping packet!
```

Use the lease specified in "... using Lease 10.0.0.23" as the <ipaddr>, and the limitation-id specified in "... with limitation-id 01:02:03" as the <limitation-id>:

```
nrcmd> dhcp limitationList 10.0.0.23 01:02:03 show
```

The result would be a list of 3 leases with the client's MAC address, the client last transaction time, and the client's host name.

## Examples

## Status

## See Also

[server](#)

## Attributes

**activity-summary-interval** [time](#) default = 60s

Sets the time that elapses between activity summary log messages when activity-summary setting is enabled in log-settings.

**addr-blocks-default-selection-tags** [nlist](#)(obj(0))

Associates a default selection tag (or list of tags) with incoming subnet-allocation requests that do not contain any subnet name data.

**addr-blocks-use-client-affinity** [bool](#) default = true

Controls whether the DHCP server allocates subnets to clients using address-blocks that the clients have already used. If you set this attribute to false, the server allocates subnets from any suitable block (based on other selection data in client messages).

**addr-blocks-use-lan-segments** [bool](#) default = false

Controls whether DHCP subnet-allocation uses the lan-segment attribute when it is configured on address-blocks.

**addr-blocks-use-selection-tags** [bool](#) default = true

Controls whether the server compares subnet name data on an incoming subnet-allocation request with the selection tag of each address-block. A block is only considered if the two match.

**cache-client-for-requests** [bool](#) default = false

Controls whether or not DHCP server will add the client to the client-cache during DHCPREQUEST (REQUEST, RENEW and REBIND in case of DHCPv6) message processing. Default, false (disable). If true, DHCP server will add the client to the client-cache during DHCPREQUEST (REQUEST, RENEW and REBIND in case of DHCPv6) message processing. If false, the DHCP server will not add the client to the

client-cache during DHCPREQUEST (REQUEST, RENEW and REBIND in case of DHCPv6) message processing. It will only add the client to the client-cache during DHCPDISCOVER (SOLICIT for DHCPv6) processing.  
Adding the client to the client-cache can improve performance when the server receives multiple messages from a client within a short period, such as would be expected for DHCPDISCOVER/DHCP OFFER, DHCPREQUEST/DHCPACK (or SOLICIT/ADVERTISE, REQUEST/REPLY) sequences.

**client-cache-count** [int](#) default = 1000

Specifies the maximum number of clients in the client cache. The DHCP server allocates the amount at startup and frees it at shutdown. If you set the value to 0, you disable client caching and the server must use persistent storage to process a DHCPREQUEST.

**client-cache-ttl** [time](#) default = 10s

Sets the maximum time-to-live in seconds for a client in cache. The DHCP server discards the entries in memory after this period.

**client-class** [bool](#) default = disabled

Controls how the DHCP server uses the client and client-class configuration objects to affect request processing. Default is false (disabled).

**client-class-lookup-id** [expr](#)

Specifies the expression used to determine a client-class solely on data contained in an incoming DHCP client request. The expression must return a string with the name of a currently configured client-class, otherwise it must return the string ''. Any return that is not a string containing the name of a currently configured client-class or '' is considered an error.

**collect-performance-statistics** [bool](#) default = false

Controls whether the DHCP server collects statistics for performance monitoring.

**collect-sample-counters** [bool](#) default = disabled

Controls whether the DHCP server collects activity statistics counters independently of the log-settings attribute flag. If false, this attribute is disabled. If true, this attribute enables collecting activity statistics.  
Note: These activity statistics counters are also enabled, if you enable 'activity-summary' logging is enabled (see 'log-settings').

**default-free-address-config** [nameref](#)(0)

Specifies the default SNMP free-address trap configuration object for the server. All scopes that are not individually configured use this default free-address value.

**defer-lease-extensions** [bool](#) default = true

Determines whether the DHCP server can defer extending leases to their full lease expiration time.  
If enabled (true), the DHCP server can defer extending the lease expiration time. Rather than give the client the full configured lease time, the server instead gives the client the remaining time on the existing lease, improving server performance and throughput. The DHCP server typically defers extending the lease if the client renews early; that is, before the client's lease renewal time (T1) elapses.  
If enabled, the DHCP server can save committing the lease to disk and updating its failover partner with an extended lease expiration time.  
If disabled, the DHCP server always attempts to extend the lease expiration time. However, there may be conditions that prevent the server from extending the lease for the full configured lease time; for example, failover protocol restrictions.

**delete-orphaned-leases** [bool](#) default = false

Controls how the DHCP server handles orphaned leases. Default is false (disabled).  
A lease is orphaned if:  
--It does not match a range or reservation in a configured scope, or  
--If it appears in the lease state database with a VPN ID that does not match a VPN in the database.  
If this attribute is true, the server deletes orphaned leases when the server next reloads. This ensures that the lease state database has no old, unconfigured leases in it, but it also may cause the total loss of lease state information due to a mistake in configuration.  
If this attribute is false, the server ignores the entries in the lease state database. If the lease was orphaned due to a configuration mistake, then when you correct the mistake, the DHCP server can use the lease.  
Whether true or false, the server cannot use the lease.

**delete-orphaned-subnets** [bool](#) default = false

Controls how the DHCP server handles information about an orphaned subnet; that is, whether it keeps the entry in its database or deletes it. Default is false (disable).

As the DHCP server starts up, it tries to locate the parent VPN and DHCP address block of each DHCP subnet. If a subnet refers to a VPN that is no longer configured, or if the server cannot locate a parent DHCP address block that contains the subnet, the server uses this attribute to decide.

#### **distribute-renewals** [bool](#) default = enabled

Controls whether the DHCP server is allowed to adjust renewals time to smooth other the server's renewal load.  
Note: If the maximum lease time option or preferred lifetime for any configured policy is set to be longer than 180 days, the server will not enable this feature.

#### **distribute-renewals-bucket-interval** [rangetime](#)(1s-2m)

Controls the time interval for the buckets used to smooth the server's load.  
If this attribute is unset, the server uses 10 seconds unless the number of buckets would exceed 100,000; in which case the server will use a time interval to limit the buckets to at most 100,000.

#### **distribute-renewals-max-renewal-time** [rangetime](#)(0-12w6d)

Controls the maximum renewal time used by the server in adjusting renewals to smooth the server's renewal load. If this attribute is unset (or 0), the server will determine this based on 50% of the maximum lease time (option 51) or preferred lifetime across all policies (named and embedded).  
This may need to be specified if policies can be supplied by external sources (such as LDAP) or if the configuration includes very long lease times or preferred lifetimes.

#### **dns-client-identity** [enumint](#)(txt=1, dhcid=2, transition-to-dhcid=3, regress-to-txt=4) default = txt

Specifies the type of resource record (RR) the server uses to identify clients in DNS updates to avoid client naming conflicts.

- 1 - txt The server will use the TXT RR in DHCPv4 DNS updates. This setting is used for backwards compatibility as pre-8.2 versions of CNR only support using TXT RRs for DHCPv4. This setting must be used if any pre-8.2 clusters are involved in doing DNS updates to the zone(s).
- 2 - dhcid The server will use the DHCID RR for DHCPv4 DNS updates. This setting should be used to support dual stack clients and can only be used if all DHCP servers doing DNS updates to the zone(s) for this configuration support and are configured to use the DHCID RR.
- 3 - transition-to-dhcid The server will use the DHCID RR for new entries in DNS and update existing entries to use the DHCID RR on the next DNS update done. This setting is required for a period of time to transition existing zones (which used TXT RRs) and this can only be done when all servers doing DNS updates have been upgraded to support use of the DHCID RR (8.2 and later). See below on how to expedite this transition.
- 4 - regress-to-txt The server will use the TXT RR for new entries in DNS and upgrade existing entries to use the DHCID RR on the next DNS update done. This setting is provided for use in cases where zones were prematurely transitioned to use the DHCID RR. See below on how to expedite this transition.

For the "transition-to-dhcid" and "regress-to-txt", it is recommend to use the following procedure to expedite the transition to the new RR type as quickly as possible:

1. Set the dns-client-identity attribute to "transition-to-dhcid" (or "regress-to-txt").
2. Note the value of the force-dns-update attribute.
3. Enable the force-dns-update attribute (set it to true), if not already enabled.
4. Reload the server.
5. After a period of time of at least the longest lease time configured in the server (for the leases in the zones being updated), set the dns-client-identity attribute to "dhcid" (or "txt") and restore the force-dns-update attribute to its earlier value if it was changed.

This attribute will provide the default value to "dns-client-identity" attribute present in DNS Update Config settings

#### **dns-timeout** [int](#) default = 15000

Controls the number of milliseconds that the DHCP server waits for a response before retrying dynamic DNS requests.

#### **docsis-version-id-missing** [string](#)

Sets the string value (maximum 255 characters) to substitute for the %@docsis-vers% variable in the policy command boot-file attribute. No default.  
DOCSIS devices must provide their version id (for example, "docsis1.0") in the vendor-class-id option. The server substitutes the DOCSIS version id as the value of the boot-file option. If the vendor class-id option is missing, or is not syntactically correct, the server uses the docsis-version-id-missing attribute for the boot-file option.  
This substitution occurs if the DHCP request packet does not contain a vendor-class-id option or the option does not contain a DOCSIS version ID.

#### **drop-old-packets** [int](#) default = 4

Sets the time, in seconds, that a packet can age and still be processed.  
The server attempts to read as many packets as possible from

the UDP input queue, and then process them quickly. If the server is very busy, it can sometimes become flooded with packets. This could delay processing some packets. In the DHCP protocol, however, some clients automatically retry packets that have not been processed in a few seconds --so allowing the server to process packets that are older than a small number of seconds. This can increase congestion without providing any real value for the clients. The drop-old-packets parameter is the number of seconds that a packet can age and still be processed by the DHCP server. If a packet is more than the value of drop-old-packets old when processed by the DHCP server, the server drops the packet.

#### **drop-packet-on-extension-failure** [bool](#) default = enabled

Causes the server to drop a packet (if possible) when it encounters an extension failure. Default is true (enable).

#### **enhanced-sample-counters** [bool](#) default = enabled

Enables collection of enhanced statistics counters by the DHCP server, which are then available with DHCP server statistics. The enhanced counters provide more detailed information, but cost the server some performance to maintain. Currently, this enables collecting milliseconds ACK/Reply latencies (instead of second based) and scope aggregation data (even if not explicitly configured).

#### **equal-priority-most-available** [bool](#) default = disabled

Controls address allocation among scopes in the same network. This attribute determines how the DHCP server allocates an address to a new client when multiple scopes have the same nonzero allocation priority. Default is false (disable). If disabled, the server uses the scope with the fewest available addresses to allocate an address to a new client (if not in a limitation list). If enabled, the server uses the scope with the most available addresses to allocate an address to a new client (if not in a limitation list). In either case, if a client is in a limitation list, among those scopes of the same priority, the one that contains other clients in the same list is always used.

#### **expression-configuration-trace-level** [int](#) default = 2

Sets the trace level to use when configuring DHCP expressions. Default is 2 (failure retry). The range is from 0 through 6, with 0 being the lowest amount of tracing and 6 the highest:

- 0 No additional tracing
- 1 No additional tracing
- 2 Failure retry
- 3 Function definitions
- 4 Function arguments
- 5 Variable lookups and literal details
- 6 Everything

Setting a high expression-configuration-trace-level imposes no performance penalty, since expressions are configured only when you restart the server.

#### **expression-trace-level** [int](#) default = 2

Sets the trace level to use when executing DHCP expressions. Optional, default 2. The range is from 0 through 6, with 0 being no tracing and 6 the highest amount of tracing:

- 0-No tracing
- 1-Failures, including those protected by (try ...)
- 2-Total failure retries (with trace level = 6 for retry)
- 3-Function calls and returns
- 4-Function arguments evaluated
- 5-Print function arguments
- 6-Datatype conversions (everything)

Setting this attribute to any level other than 0, 1, or 2 imposes a considerable performance penalty. The setting of 1 only traces when there is a failure in an expression. The default setting of 2 re-executes evaluating an expression that fails at the outermost level with the expression-trace-level=10 for the duration of the re-execution, to provide maximum debugging assistance. Note: This attribute can be updated dynamically and does not require a DHCP server reload.

#### **extension-trace-level** [int](#) default = 0

Sets the value of the extension callback trace logging level, which can be useful in developing and debugging extensions. Default is 0. The range is from 0 through 4, with more tracing at higher levels (higher levels also included tracing at lower levels). The levels are approximately as follows, depending on the language the extension is implemented in:

- 0 - All: No tracing
- 1 - DEX: log() callback with invalid eLevel traced  
TCL: Some callbacks with invalid arguments traced
- 2 - All: Not used (same as 1)
- 3 - DEX: put/putBytes failures traced  
TCL: Unknown methods, wrong number of arguments, and put/putBytes callbacks traced
- 4 - All: Callbacks traced (success with results or failure)

Note: This attribute can be updated dynamically and does not require a DHCP server reload.

#### **failover-recover** [date](#) default = none

With failover enabled, sets the time at which the server is initialized and goes into RECOVER state. No default. If failover-recover is non-zero, and the server has no record of a previous failover state in its stable storage,



then the server assumes that it was previously operational but lost all of its stable storage. The time value in the failover-recover parameter represents a time equal to or later than the last known time the server could have been operational. The server attempts to refresh its information from its failover partner by entering RECOVER state and requesting a complete update of all binding information. Time values for this attribute can be in hours (for example, -2h two hours ago) or month day hour:minute[:second] year. Use the local time of the nrcmd process. Formats for the date are:

```
-
  where is a decimal number and is one of 's', 'm',
  'h', 'd', 'w', in which 's' is seconds, 'm' is minutes, 'h' is
  hours, 'd' is days and 'w' is weeks.
  [:]
  where is the name or first three letters of the name of
  the month, is the hour on a 24- hour clock, and is
  the fully-specified year or a two-digit representation in which
  98 = 1998, 99 = 1999 and all other two digit values XX = 20XX.
```

#### **force-dns-updates** [bool](#) default = false

Controls whether the DHCP server retries a DNS update whenever a client renews its lease, even if it appears to the server that the update was already completed successfully. Default is false (disable). This attribute uses one of the following values:

- forward DnsUpdateConfig object (if configured)
- reverse DnsUpdateConfig object (if configured)
- the default (or where appropriate the server configured value or default value).

#### **get-subnet-mask-from-policy** [bool](#) default = false

Controls whether the DHCP server searches all relevant policies for a subnet mask option to construct a response to a client. Default is false (disable). Normally, the DHCP server retains the subnet mask configured in the scope containing the base being granted to the DHCP client.

#### **gss-tsig-config** [nameref\(0\)](#)

Identifies the gss-tsig configuration object used by DHCP server.

#### **ha-dns-failover-timeout** [rangeint\(30-120\)](#) default = 30

Sets the maximum time period in seconds that the DHCP server waits for replies from a DNS server before failing over to its partner. Default is 30s. To use this attribute, first configure your DHCP server to perform HA DNS updates. If DHCP is configured for HA-DNS updates, dns-timeout parameters are set to fit max-dns-retries within ha-dns-failover-timeout. For example, with a default ha-dns-failover-timeout value of 30 seconds and max-dns-retries of 3, dns-timeout is set to 6 seconds, so that after 3 retry attempts and an ha-dns-failover-timeout, the DHCP server failovers. The DHCP server uses a minimum limit value for a dns-timeout of 2 seconds and dns-retries of 1.

#### **hardware-unicast** [bool](#) default = enabled

Controls whether the DHCP server sends unicast rather than broadcast responses when a client indicates that it can accept a unicast.

#### **ignore-cisco-options** [nlist\(obj\(0\)\)](#)

Skips processing the Cisco-specific DHCP options specified by name in the comma-separated list. No default. Allowable option names are vpn-id (185), cisco-vpn-id (221), and subnet-alloc (220). Use this attribute only if clients use the options for other purposes.

#### **ignore-icmp-errors** [bool](#) default = enabled

Controls how the DHCP server handles ICMP ECHO (ping-before-offer) requests. Default is true (enable). If you enable this attribute and configure the DHCP server to send ICMP ECHO requests, the server makes unavailable any address for which it receives an ECHO reply within its configured timeout period. If you disable this attribute, the DHCP server also treats ICMP DEST UNREACHABLE and TTL EXPIRED error messages that it receives after sending ICMP ECHO requests as grounds for making an address unavailable.

#### **ignore-requests-for-other-servers** [bool](#)

Controls whether to prevent the normal DHCP server response to client requests for other servers. If not explicitly set, the default is true if failover is in use and false otherwise.

It is recommended not to set this to false for failover if either failover partner has more than one interface over which DHCP packets may be received.

If false and the DHCP server detects a client requesting a lease from another server for an address that this server is configured to control, it sets the lease to unavailable. Some clients, however, might send request packets with bad server ID options -- rather than packets actually directed to other servers -- that the server wrongly interprets as an unavailable address.



If true, the DHCP server ignores requested not addressed to it.

#### **import-mode** [bool](#) default = disabled

Controls whether the DHCP server recognizes only packets generated from the import leases command, ignoring all others. Default is false (disable). Use this attribute to update your DHCP server, preventing clients from receiving addresses during this period.

#### **inhibit-busy-optimization** [bool](#) default = false

Controls whether the server uses optimization to recover from periods of congestion. Default false (disable). By default, the DHCP server determines that it is heavily loaded when the number of request packets reaches two-thirds of the total allocated. The server logs a message and attempts to recover from the congestion by performing several optimizations. For example, it relaxes the requirement to keep the client's last transaction time updated to the granularity specified by the last-transaction-time-granularity attribute. When the number of request packets drops to one-third of the total allocated, the server logs a message and returns to normal operation. If you enable the inhibit-busy-optimization attribute, the server does not use the optimizations or log the messages when it gets congested.

#### **initial-environment-dictionary** [string](#)

Contains attribute-value pairs that initialize all environment dictionaries used within the DHCP server. Use these attribute-value pairs to configure extensions or expressions without having to re-write the executable code in the extension or expression. The string must have the format:  
"attribute1=value1,attribute2=value2, ... ,attributen=valuen"

#### **ip-history** [enumby](#) default = disabled

Enables the lease history database for DHCPv4, DHCPv6, or both.

0 disabled	No lease history is recorded. Default.
1 v4-only	The server records lease history for DHCPv4 leases only.
2 v6-only	The server records lease history for DHCPv6 leases only.
3 both	The server records lease history for both DHCPv4 and DHCPv6 leases.

#### **ip-history-max-age** [time](#) default = 4w

If ip-history is enabled, determines how long IP records are kept in the database. Default is 4w. The server accumulates database records over time as lease bindings change. The ip-history-max-age attribute establishes a limit on the age of the history records kept in the database. The server periodically examines the lease history records, establishes an age threshold based on this parameter, and deletes any records that represent bindings that end before the threshold. The history records are trimmed by default once a day, at 3:00 a.m. local time.

#### **last-transaction-time-granularity** [time](#) default = 1w

Sets the time, in seconds, to guarantee how accurate to keep the last transaction time (in the lease DB) when the lease is not otherwise being written to the lease DB. Default is 1 week. Do not set this lower than 60 seconds. For optimal performance, set it to a value that is greater than half of your lease time. The server can maintain an accurate record of the time it last interacted with a DHCP client concerning a given lease. This setting provides the control over how accurate that time is guaranteed to be. A setting of 300 seconds, for instance, would allow the server to avoid database updates whose sole purpose is to update a last transaction time that is less than 5 minutes in the past.  
Note: This attribute is not used if defer-lease-extensions is disabled.

#### **ldap-mode** [enumint](#)(round-robin=1, failover=2) default = 1

Determines the preference for using LDAP servers if multiple LDAP servers are configured. This attribute has two possible values:

1 round-robin	- The DHCP server ignores LDAP server preferences. It treats all LDAP servers (those configured to handle client queries and those configured to accept lease-state updates) equally.
2 failover	- The DHCP server uses the active LDAP server with the lowest preference. If the preferred server loses its connection or fails, the DHCP server uses the next LDAP server in preference order. The DHCP server uses servers with equal preference in round-robin order.

#### **lease-retention-max-age** [time](#) default = 0

This enables lease time restrictions and specifies the longest time, in the past (from the current time), to which times in a lease are restricted. This can be used to meet data retention restrictions for privacy protection. If not specified, no restrictions are placed on how far back in time the times associated with a lease may be. In order for lease retention limitation to take place for a lease, not only does the lease-retention-max-age need to be non-zero, but the individual lease itself must fall under a policy where the lease-retention-limit attribute is set in that policy. This value, if configured, must be greater than 8 hours. If it is configured as non-zero and less than 8 hours, it will be set

to 8 hours.

#### lease-retention-min-age [time](#)

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to retention limitation (see lease-retention-limit in the policy) will not be allowed to grow older than lease-retention-max-age. As these times progress toward lease-retention-max-age in the past, they will periodically be reset to lease-retention-min-age in the past. This value must be at least 6 hours less than the lease-retention-max-age; if not or not specified, then lease-retention-max-age minus 6 hours is used. Periodically interacting with leases in order to keep their older times between lease-retention-min-age and lease-retention-max-age involves some processing, and the closer these two values are together the more frequently this processing must take place. This is regardless of the absolute values of these attributes! Therefore, setting the lease-retention-min-age one to several days before the lease-retention-max-age would minimize the additional server processing devoted to lease-retention limitation. Ignored unless the lease-retention-max-age is specified.

#### log-format [flags](#)(header-in-packet-detail=1) default =

Controls how the DHCP server logs certain data to the log files. Possible flags are:

- header-in-packet-detail
  - Controls the format of packet-detail logging. If unset, the server uses a new, higher performance format, if such logging is enabled. The new format does not include the log line header for each line, and will not intermix packet detail with other log messages. (This is the same format used by the DNS server for packet-detail logging.)
  - If set, the traditional format for DHCP packet formatting is used, which includes the log line header for each line and could intermix packet detail for different packets and other log messages.

The new format is highly recommended, except where applications that parse this information require the traditional format.

#### log-settings [flags](#)(default=1, incoming-packets=2, missing-options=3, incoming-packet-detail=4, outgoing-packet-detail=5, unknown-criteria=6, dns-update-detail=7, client-detail=8, client-criteria-processing=9, failover-detail=10, ldap-query-detail=11, ldap-update-detail=12, ldap-create-detail=13, leasequery=14, dropped-waiting-packets=15, no-success-messages=16, no-dropped-dhcp-packets=17, no-dropped-bootp-packets=18, no-failover-activity=19, activity-summary=20, no-invalid-packets=21, no-reduce-logging-when-busy=22, no-timeouts=23, minimal-config-info=24, no-failover-conflict=25, deprecated=26, atul-detail=26, v6-lease-detail=27) default = default,incoming-packets,missing-options,activity-summary

Determines which events to log in the log files. Default flags are default, incoming-packets, and missing-options. Logging additional detail about events can help analyze a problem. However, leaving detailed logging enabled for a long period can fill up the log files. Possible flags are:

- activity-summary
  - This setting will cause a summary message to appear every 1 minute. It is useful when many of the no-xxx log settings are enabled, to give some idea of the activity in the server without imposing the load required for a log message corresponding to each DHCP message. The time period for these messages can be configured with the DHCP server property activity-summary-interval.
- client-criteria-processing
  - This setting will cause a log message to be output whenever a scope is examined to find an available lease or whenever a scope is examined to determine if a lease is still acceptable for a client who already has one. It can be very useful when configuring or debugging client-class scope criteria processing. It causes moderate amount of information to be logged and should not be left enabled as a matter of course.
- client-detail
  - This setting will cause a single line to be logged at the conclusion of every client-class client lookup operation. This line will show the composite of the data found for the client as well as the data that found in the client's client-class. It is useful when setting up a client-class configuration and for debugging problems in client-class processing.
- default
  - The default gives a low level of logging in several parts of the DHCP server. If you unconfigure the default, even this logging will not appear.
- dns-update-detail
  - This setting causes the server to log a message as it sends each dns update and as it receives replies to update messages.
- dropped-waiting-packets
  - If the value of max-waiting-packets is non-zero packets may be dropped if the queue length for any IP address exceeds the value of max-waiting-packets. If dropped-waiting-packets is set, the server will log a message whenever it drops a waiting packet from the queue for an IP address.
- failover-detail
  - This setting causes the server to log a single message for most failover transactions. The information logged is very useful for understanding how failover is operating, and should be included if at all possible when sending requests for support regarding failover issues.
- incoming-packet-detail
  - This setting will cause the contents of every DHCP packet received by the DHCP server to be interpreted in a human readable way and printed in the log file. This enables the built-in DHCP packet sniffer for input packets. The log files will fill up (and turn over) very rapidly when this setting is enabled. This setting also causes a significant performance impact on the DHCP server and should not be left enabled as a matter of course.
- incoming-packets
  - This setting (on by default) will cause a single line message to be logged for every incoming packet. This is especially useful when initially configuring a DHCP server or a BOOTP relay, in that an immediate positive indication exists that the DHCP server is receiving packets.
- ldap-create-detail

This setting will cause log messages to appear whenever the dhcp server initiates an lease state entry create or delete to LDAP server, receives response and retrieves result or error messages.

**ldap-query-detail**  
This setting will cause log messages to appear whenever the dhcp server initiates a query to LDAP server, receives response and retrieves result or error messages.

**ldap-update-detail**  
This setting will cause log messages to appear whenever the dhcp server initiates an update lease state to LDAP server, receives response and retrieves result or error messages.

**leasequery**  
This setting will cause log messages to appear when leasequery packets are processed without internal errors and result in an ACK or a NAK.

**minimal-config-info**  
This setting will reduce the number of configuration messages printed when the server starts or reloads. In particular, it will not log a message for every scope.

**missing-options**  
This setting (on by default) will cause a message to be logged whenever an option requested by a DHCP client has not been configured in a policy and therefore cannot be supplied by the DHCP server.

**no-dropped-bootp-packets**  
This setting will cause the single line message normally logged for every BOOTP packet that is dropped to not appear.

**no-dropped-dhcp-packets**  
This setting will cause a single line message normally logged for every DHCP packet that is dropped due to DHCP configuration to not appear. (See no-invalid-packets for messages associated with packets dropped because they are invalid.)

**no-failover-activity**  
This setting will cause normal activity and some warning messages logged for failover to not appear. Serious error log messages will continue to appear independent of this log-setting.

**no-failover-conflict**  
This setting will cause conflicts between failover partners to not be logged.

**no-invalid-packets**  
This setting will cause a single line message normally logged for every DHCP packet that is dropped due being invalid to not appear. (See no-dropped-dhcp-packets for messages associated with packets dropped due to DHCP server configuration.)

**no-reduce-logging-when-busy**  
Normally, the DHCP server will reduce logging when it becomes very busy (i.e., when it has used over 2/3 of the available receive buffers (itself a configurable value)). It will set no-success-messages, no-dropped-dhcp-packets, no-dropped-bootp-packets, no-failover-activity, no-invalid-packets, and clear everything else except activity-summary. If no-reduce-logging-activity is set, then the server will not do this. It will restore the previous settings when the server becomes unbusy (i.e., when it has used only 1/3 of the available receive buffers).

**no-success-messages**  
This setting will cause the single line message that is normally logged for every successful outgoing DHCP response packet to not appear. It affects logging only for successful outgoing DHCP response packets.

**no-timeouts**  
This setting will cause messages associated with timeout of leases or offers not to appear in the log file.

**outgoing-packet-detail**  
This setting will cause the contents of every DHCP packet transmitted by the DHCP server to be interpreted in a human readable way and printed in the log file. This enables the built-in DHCP packet sniffer for output packets. The log files will fill up (and turn over) very rapidly when this setting is enabled. This setting also causes a significant performance impact on the DHCP server and should not be left enabled as a matter of course.

**unknown-criteria**  
This setting will cause a single line log message to appear whenever a client entry is found which specifies selection criteria that is not found in any scope appropriate for that client's current network location.

**v6-lease-detail**  
This setting causes the server to log individual messages regarding DHCPv6 leasing activity (in addition to or in place of a single message per client transaction depending on no-success-messages, or client timeout event depending on no-timeouts).

Note: This attribute can be updated dynamically and does not require a DHCP server reload.

**mac-address-only** [bool](#) default = disabled

Controls whether the DHCP server uses the client's MAC address as the only client identifier. The standard behavior, as specified in RFC 2132, is to use the client-id option (if it is present) as the unique client identifier. Default is false (disable).

CAUTION: Use this attribute with care. When enabled, it precludes a MAC address from getting multiple IP addresses per network. It forces the server to use a Client-Identifier (CID) created from the MAC address instead of the RFC described client-id contained in the request. This can preclude newer devices that take multiple IP addresses. Enabling, or later disabling, this attribute can also have an operational impact. Clients that originally obtain addresses through a client-id cannot renew them once they are assigned attributes based on a MAC address.

**map-radius-class** [enumint](#)(none=0, map-as-tag=1, map-as-class=2, append-to-tags=3) default = 0

Controls how to map the radius attribute, if present, in the client request relay-agent option:

0	none	Ignores the radius class name default
1	map-as-tag	Maps the radius class to selection-tags
2	map-as-class	Maps the radius class directly to a client-class name

3 append-to-tags Appends the radius class to the selection-tags

**map-radius-pool-name** [enumint](#)(none=0, map-as-tag=1, map-as-class=2, append-to-tags=3) default = 0

Controls use of the radius framed-pool attribute, if present, in a client relay-agent option.

0	none	Ignores the framed-pool attribute
1	map-as-tag	Maps the framed-pool attribute to selection-tags
2	map-as-class	Maps framed-pool attribute directly to a client-class name
3	append-to-tags	Appends the user-class-id to the selection-tags

**map-user-class-id** [enumint](#)(none=0, map-as-tag=1, map-as-class=2, append-to-tags=3) default = 0

Controls how the server uses the user-class-id option. Values are:

0	none	Ignores the user class-id(default)
1	map-as-tag	Maps the user-class-id to selection-tags
2	map-as-class	Maps user-class-id directly to a client-class name
3	append-to-tags	Appends the user-class-id to the selection-tags

**max-client-leases** [rangeint](#)(1-65535) default = 50

Sets the maximum number of leases, regardless of state or whether reserved or not, that the server can associate with a DHCPv6 client. A DHCPv6 lease is always associated with a client; if it is not, it is deleted. This setting is to prevent a client from using lots of leases (such as by issuing many requests with different IAID values). It is not intended to limit the number of active leases a client may have. Note that leases in REVOKED state (generally used to handle reconfiguration events) are excluded from this limit. This limit is not applied when existing leases are loaded from the lease state database during server start-up.

**max-dhcp-requests** [int](#) default = 500

Controls the number of buffers that the DHCP server allocates for receiving client requests. When you enable failover, allocate at least 150 buffers. Up to 1500 buffers might be reasonable for high capacity installations. Caution: Increasing the number of buffers can degrade performance. Cisco recommends using the default value in most situations. When buffer size exceeds capacity, a burst of DHCP activity can clog the server with requests that become stale before they are processed. This increases the processing load and might severely degrade performance as clients try to obtain a new lease. A lower buffer setting throttles requests and avoids wasted processing on requests that would otherwise be stale. When using LDAP client lookups, buffers should not exceed the LDAP lookup queue size defined by the total number of LDAP connections and the maximum number of requests allowed for each connection. Set the LDAP queue size to match the LDAP server's capacity to service client lookups.

**max-dhcp-responses** [int](#) default = 1000

Controls the number of buffers that the DHCP server allocates for responses to client requests. The server ignores this value if max-dhcp-requests is higher, or other configuration options such as failover require that the value be set higher than configured.

**max-dns-renaming-retries** [int](#) default = 3

Controls the number of times that the DHCP server can attempt adding a host into DNS, even if the DHCP server detects that the hostname is already present in DNS. The DHCP server attempts to modify the hostname in order to resolve a conflict on each failed update.

**max-dns-retries** [int](#) default = 3

Controls the number of times that the DHCP server can try to send dynamic updates to a DNS server.

**max-dns-ttl** [int](#) default = 86400

Sets the time to live (TTL) ceiling, in seconds, for DNS records added through dynamic updates. When the DHCP server adds a DNS record, it uses a TTL of the minimum of either this ceiling or one third the lease time.

**max-ping-packets** [int](#) default = 500

Sets the number of buffers the server allocated for sending and receiving ICMP ping messages. See the 'ping-clients' and scope 'ping-clients' attribute.

**max-waiting-packets** [rangeint](#)(1-10) default = 6

Sets the maximum number of packets that can wait for a particular IP address. The server queues only the most recently received n packets (of an address) for processing. If an additional packet associated with that address arrives and n packets are already queued, the server drops the oldest packet and queues the new one. It also drops duplicate packets (whose XID, client ID, and

MAC address are the same as one already queued).  
Dropped packets are logged if the log setting  
dropped-waiting-packets is set. It is off by default.

**min-dns-ttl** [time](#) default = 10m

Minimum value for time to live (TTL) in seconds, for DNS records added through dynamic updates. When the DHCP server adds a DNS record, the TTL value will be min-dns-ttl if one third the lease time is less than the min-dns-ttl value.

**multicast-addresses** [nlist\(obj\(0\)\)](#) default = ff02::1:2,ff05::1:3

Controls the default multicast addresses that are enabled on interfaces. The address ff02::1:2 is required if any DHCPv6 clients are directly connected to the link associated with an interface. The address ff05::1:3 is the default multicast address that relay agents use to relay DHCPv6 requests.

**one-lease-per-client** [enumint](#)(disabled=0, last-client-preferred=1, first-client-preferred=2) default = disabled

Controls whether the DHCP server releases other leases a client might have on other LAN segments on this server.

- 0 disabled
- 1 last-client-preferred
- 2 first-client-preferred

Within one LAN segment, the DHCP server never allocates more than one DHCPv4 address to a single client. Across multiple independent network LAN segments, however, a single client might have one address allocated on several networks.

In an enterprise environment this might happen when a laptop user travels from building to building, causing no particular problems. In a service provider context, this might happen when an unapproved user attempts to clone the MAC address of a legitimate subscriber, causing a theft of service.

The one-lease-per-client feature limits a single client to one lease over all of the networks configured on the DHCP server. This limit only affects a client's ability to have concurrent leases to different IP addresses on more than one network at a time on the DHCP server.

In the last-client-preferred approach, the client with the most recent lease is given preference, and any other leases held by the same client are released.

In the first-client-preferred approach, the first lease that a particular client receives from the DHCP server is the only lease that client is allowed. In the event that the client moves, the first lease must be made available (perhaps by a force-available command or by letting it expire) before the client is allowed to lease another IP address on a different network.

Caution: Use the first-client-preferred approach with great care, since manual intervention might be required to ensure proper operation.

Both forms of one-lease-per-client require additional bookkeeping overhead beyond that normally done in the DHCP server.

**ping-clients** [bool](#) default = disabled

Controls the default value of the Scope's 'ping-clients' attribute if not explicitly configured on a scope. This attribute allows the administrator to control the server wide default for pinging an address before assigning it to a client. It can be explicitly overwritten for each scope. If enabled, see the 'ping-timeout' attribute as it may also need to be set.

**ping-timeout** [rangeint](#)(0-10000) default = 300

Sets the server default for the number of milliseconds the server waits for ping responses. If you make this value too large, you slow down the lease offering processes. If you make this value too small, you reduce the effectiveness of pinging addresses before offering them. 300 milliseconds (the default value) is often the best choice.

When ping-clients is enabled (either on a scope or server-wide), this value is used as the default if not explicitly configured on a scope.

**priority-address-allocation** [bool](#) default = disabled

Controls address allocation among scopes in the same network and within an individual scope as well.

When enabled, any scope without an explicit setting for allocation-priority is configured with an allocation-priority equal to the network number of the scope. Similarly, any scope without an explicit setting for allocate-first-available is considered enabled.

Explicit settings for either of these scope attributes override the priority-address-allocation set for that scope.

This attribute gives the administrator a way to change address allocation in a server wide manner without having to separately configure each scope.

**return-client-fqdn-if-asked** [bool](#) default = true

Controls whether the system returns the client FQDN option in the outgoing packet to the client if the incoming packet contains the FQDN option. Default false (disable).

If true (enable), the option flags are always set to 0x3 and the RCODE1 and RCODE2 to 255. Any string contained in the incoming packet is returned, even if the use-client-fqdn attribute is disabled and regardless of the actual FQDN.

**save-vendor-class-id** [bool](#) default = false

Determines whether to store the vendor-class-id, as

furnished in a dhcp request option 60, as part of the lease record.

**server-flags** [flags](#)(serial-lease-loading=1, save-vendor-specific-info=2, rfc3315-advertise=3, rfc3315-t1t2=4, compress-lease-records=5, ignore-director-bypass=6, no-pre-load=7) default =

Defines flags that control DHCP server behavior.

**Compress-lease-records**

Determines whether lease records are compressed. Compressing lease records cost a bit of CPU time (mostly for compression, but also for decompression) but should reduce the disk load to the lease state database as more lease records should fit into a database page. This only impacts FUTURE writes to lease records; a record will be written either compressed (enabled) or not compressed (disabled).

Warning: Enabling this setting makes the lease state database incompatible with pre-9.0 releases.

**Ignore-director-bypass**

Determines whether the server should ignore DHCPv6 Cisco Enterprise-ID (9) Vendor-specific Information option (17) vendor option 22. If not set to ignore and this vendor-specific Information option is present (in the outermost Relay-Forw), the server will send the response back to this Relay-Forw's peer-address rather than the packet's source, bypassing this outermost Relay-Forw.

**No-pre-load**

Determines whether the server should issue a copy of the lease state database to the null device on first start to load the file system cache with the database. This is to expedite loading of leases. If not set, the database is copied. If set, the database is not copied. Warning: This assumes that there is sufficient "free" memory available for the dhcp.ndb file to be loaded into the file system cache, which should normally be the case for best performance.

**RFC3315-advertise**

Determines whether to use RFC 3315 or RFC 7550 formatted Advertise responses when there are no addresses leases available. When set, RFC 3315 formatted Advertise responses are generated.

**RFC3315-t1t2**

Determines whether to use RFC 3315 or RFC 7550 T1/T2 times in responses. For RFC 7550, the T1/T2 times are set to the same values for all IAs, not just on a per IA basis as for RFC 3315. When set, RFC 3315 T1/T2 times are returned in responses.

**Save-vendor-specific-info**

Determines whether to store the vendor specific information as furnished in a dhcp request option 43, as part of the lease record.

**Serial-lease-loading**

Load the leases serially, rather than loading the leases in parallel (using more than one thread). While parallel loading is generally faster, it can also cause some issues with the GLIBC MALLOC implementation on Linux.

**skip-client-lookup** [bool](#) default = false

Determines whether the DHCP server looks up the client entry in the database to do client-class processing. This value can be examined as well as changed in a script at the pre-client-lookup extension point. Default false (disable). If true, the server skips the client entry.

**sms-lease-interval** [int](#) default = 1100

Sets the amount of time, in milliseconds, that the DHCP server waits between sending addresses to the System Management Server (SMS) when executing updateSms command.

**sms-library-path** [string](#)

Overrides the internal default value for the SMS.dll. Default is an empty string. If the string is empty, the system defaults to the internal server default of smsrsgen.dll. If the string is not empty, its value overrides the internal SMS library name, smsrsgen.dll. If the system path does not include the location of the SMS dll, you should provide the absolute path of the dll.

**sms-network-discovery** [int](#) default = 0

Determines whether the DHCP server generates SMS network discovery records. If this attribute is set to 0, you disable SMS network discovery. If it is set to 1, you enable discovery. Use this attribute in conjunction with the dhcp updateSMS command.

**sms-site-code** [string](#)

Specifies the site code name of the SMS server that receives discovery records when you use the updateSMS keyword. For proper functioning, make sure that you initialize this attribute to the appropriate site code. The default value is an empty string, but this prohibits data discovery to complete successfully. So, you must provide the site code.

**synthesize-reverse-zone** [bool](#) default = enabled

Controls whether the DHCP server automatically generates the name of the reverse zone (in-addr.arpa or ip6.arpa) that receives PTR records updates. Default is true (enable). If true, you are not required to configure an explicit dns-reverse-zone-name in the DNS update configuration. Instead, the DHCP server uses the IP address of each lease and the dns-host-bytes



attribute of the scope or the reverse-zone-prefix-length attribute of the prefix to synthesize the reverse zone name for each update. The server trims the specified host bytes from the low-order bytes of the address, and turns the remaining bytes into a zone name of the form '.b.a.in-addr.arpa.'. The host-bytes low-order bytes of the address are used to form the hostname within that zone. Similarly, the prefix length is used to form the 'ip6.arpa' zone name, and the low-order bytes are used to form the hostname.

**traps-enabled** [flags](#)(all=1, server-start=2, server-stop=3, free-address-low=4, free-address-high=5, dns-queue-size=6, other-server-down=7, other-server-up=8, duplicate-address=9, address-conflict=10, failover-config-error=11, free-address6-low=12, free-address6-high=13, duplicate-address6=14, duplicate-prefix6=15, address6-conflict=16, prefix6-conflict=17) default =

Determines the traps that this server is configured to send.

- 1 all  
Sends notifications for all server events.
- 2 server-start  
Sends notifications whenever the server is started or reinitialized.
- 3 server-stop  
Sends notifications whenever the server is stopped.
- 4 free-address-low  
Sends notifications when the number of free IP addresses becomes less than or equal to the low threshold.
- 5 free-address-high  
Sends notifications when the number of free IP addresses exceeds the high threshold after having previously triggered the free-address-low trap.
- 6 dns-queue-size  
Sends notifications when the DHCP server's DNS queue fills and the DHCP server stops processing requests.
- 7 other-server-down  
Sends notifications when another server (DHCP, DNS, or LDAP) stops responding to this DHCP server.
- 8 other-server-up  
Sends notifications when another server (DHCP, DNS, or LDAP) responds after having been unresponsive.
- 9 duplicate-address  
Sends notifications whenever a duplicate IP address is detected.
- 10 address-conflict  
Sends notifications when an address conflict with another DHCP server is detected.
- 11 failover-config-error  
Notifies when a configuration mismatch between DHCP failover partners occurs.
- 12 free-address6-low  
Sends notifications when the number of free IPv6 addresses becomes less than or equal to the low threshold.
- 13 free-address6-high  
Sends notifications when the number of free IPv6 addresses exceeds the high threshold after having previously triggered the free-address6-low trap.
- 14 duplicate-address6  
Sends notifications whenever a duplicate IPv6 address is detected.
- 15 duplicate-prefix6  
Sends notifications whenever a duplicate prefix is detected. This trap is deprecated.
- 16 address6-conflict  
Sends notifications when an IPv6 address is detected to be leased to different clients by the failover partners.
- 17 prefix6-conflict  
Sends notifications when an IPv6 delegated prefix is detected to be leased to different clients by the failover partners.

**trim-host-name** [bool](#) default = enabled

Controls whether or not DHCP server trims the hostname string to the first period (or dot) character (used to update DDNS records and to return host-name-option to clients). Default, true (enable).  
If true, the hostname is truncated before the period.  
If false, the DHCP server retains the period characters in the hostname.

**update-dns-for-bootp** [bool](#) default = enabled

If the server is replying to a BOOTP request, and is offering a lease from a Scope which is configured to perform DNS updates, it will check this property before beginning the DNS update. This feature allows an administrator to prevent DNS updates for BOOTP clients, while allowing updates for DHCP clients.

**upgrade-unavailable-timeout** [time](#) default = 24h

Controls the time given to a lease in the database that has no expiration; that is, it became unavailable prior to installing Network Registrar. The DHCP server uses the upgrade-unavailable-timeout for the expiration time of the unavailable lease. Default is 86400 seconds (1 day).

**use-client-fqdn** [bool](#) default = true

Controls whether the server examines the client-FQDN option for the hostname. Default is true (enable).  
If true, the server ignores any characters after the first dot (.) because the domain is determined from the scope.  
If false, the server does not determine the hostname using this. This is useful if the client is sending unexpected or junk characters.

**use-client-fqdn-first** [bool](#) default = true

Controls whether the DHCP server looks at the client-FQDN option on incoming packets first, before looking at the 'hostname' option. Default is true (enable).

If true and the client-FQDN option specifies a hostname, the server uses that hostname.  
If the client-FQDN option is not present in the incoming packet, the server uses the hostname from the 'hostname' option.  
If false, the server also uses the hostname from the 'hostname' option.

**use-dns-update-prereqs** [bool](#) default = true

Controls whether the DHCP server adds prerequisites to DNS update messages for DHCPv4 leases. Default true (enable).  
If true, the DHCP server includes prerequisites in DNS update messages to make sure the client is using the domain name, before it is updated with the current lease (IP address). For DHCPv6 updates, prerequisites are always used.  
If false, the DHCP server assumes the requesting client is entitled to the domain name. In this case, it does not include prerequisites in the DNS update message, and associates the client lease with that domain name.  
Note: The DHCP server always adds prerequisites to the DNS update message while adding a new domain name record on behalf of a client acquiring a new lease and removing the domain name record when a client releases its lease or the lease expires.

**use-host-name** [bool](#) default = true

Specifies whether the server looks at the 'hostname' option for the hostname. Default is true (enable).  
If true, the server obtains the hostname from the 'hostname' option.  
If false, the server does not obtain the hostname from this option. This is useful if the client is sending unexpected or junk characters.

**use-ldap-client-data** [bool](#) default = disabled

Controls whether the DHCP server attempts to read client-entry data using the configuration supplied by the 'ldap' command. Default is false (disable).

**v6-client-class-lookup-id** [expr](#)

Defines the expression used to assign a client-class based solely on data contained in an incoming DHCPv6 client request. No default.  
The expression must return a string that is the name of a currently configured client-class; otherwise, the expression must return the string ''. Any return that is not a string containing the name of a currently configured client-class or '' is considered an error.

**v6-default-free-address-config** [nameref\(0\)](#)

Specifies the default SNMP v6 free-address trap configuration object for the server. All Prefixes and Links that are not individually configured use this default free-address value.

**validate-client-name-as-mac** [bool](#) default = false

Controls whether client names must be formatted as a MAC address (or the literal string 'default') when a client is created.  
Setting this attribute to true is not recommended as creating DHCPv6 clients requires this attribute be false (the default).  
If true, when creating a new client the name must be one of the following:  
- A mac address (,), typically 1,6,xx:xx:xx:xx:xx:xx  
- A blob formatted value, such as xx:xx:xx:xx:xx:xx which which will be converted to the mac address format  
- The string 'default'  
Any other name will return an error.  
Note that this blob format conversion only applies when creating a client and does not occur on client lookups.  
If false, any name is allowed and is used exactly as specified. The client name must match the lookup keys generated from the client-lookup-id (or v6-client-lookup-id) expression.

**vpn-communication** [bool](#) default = true

Controls the ability of the DHCP server to communicate with clients that are on a different VPN from the server. Default is true (enable).  
If true, the server communicates with DHCP clients residing on a different VPN by using an enhanced DHCP Relay Agent capability. This enhanced DHCP Relay Agent capability is indicated by the appearance of the server-id-override sub-option in the relay agent information option (Option 82).

## dhcp-address-block

dhcp-address-block - Defines a contiguous range of IP address space from which the DHCP server may allocate subnets

### Synopsis

```
dhcp-address-block <name> create <address> [<attribute>=<value> ...]
dhcp-address-block <name> delete
dhcp-address-block list
dhcp-address-block listnames
```



```

dhcp-address-block listbrief
dhcp-address-block <name> show
dhcp-address-block <name> listsubnets

dhcp-address-block <name> get <attribute>
dhcp-address-block <name> set<attribute>=<value> [<attribute>=<value> ...]
dhcp-address-block <name> unset <attribute>
dhcp-address-block <name> enable <attribute>
dhcp-address-block <name> disable <attribute>

```

## Description

The `dhcp-address-block` command creates and sets attributes for Network Registrar DHCP address blocks. The command applies only to address block objects that are designated in the DHCP server for subnet allocation to clients. When a DHCP server receives a request to allocate a subnet to a client, it does so by subdividing its available address-blocks. In this context, a DHCP address block is a contiguous range of IP address space that is delegated to the DHCP server for assignment. The DHCP server expects to subdivide these DHCP address blocks for delegation to some other server or device, or for its own use in interaction with DHCP clients.

DHCP address blocks can parent one or more subnets. Subnets are also contiguous ranges of IP address space that are bound to a specific client, usually a router or another DHCP server. DHCP address blocks and subnets are similar to scopes in that they contain address ranges and other attributes necessary to configure the DHCP client-server interaction. Unlike scopes, DHCP address blocks and subnets do not have address ranges available for assignment to DHCP clients and do not contain reserved addresses.

In a virtual private network (VPN) deployment where multiple VPNs use the same private address space, you can use logically identical DHCP address blocks simultaneously on multiple VPNs.

## Examples

```

nrcmd> dhcp-address-block example1 create 10.10.0.0/16
nrcmd> dhcp-address-block example1 set policy=p2
nrcmd> dhcp-address-block example1 show

```

## Status

## See Also

[dhcp-subnet](#)

```

vpn [] (AT STRING, Optional, default: <none>)
  This is a "virtual" property. Use this property to set or
  get the vpn-id by vpn name instead of by id.

```

## Attributes

**address** [subnet](#) required

Determines the IP subnet address (in the address/mask format) of a DHCP address block. This value is defined when you create the address block.

**default-subnet-size** [int](#) default = 28

Sets the default subnet size for allocations from this address block.

**deprecated** [bool](#) default = false

Determines whether the server deactivates a DHCP address block. The server ignores a deprecated DHCP address block for new subnet allocations. It allows existing clients to renew their subnets, but indicates to them that the subnet is deprecated. The client then prepares to release the deprecated subnet or subnets back to the server.

### embedded-policy [obj\(0\)](#)

Displays the embedded policy object for this DHCP address block. Read-only. Use the `dhcp-address-block-policy` command to set the embedded policy.

### name [string](#)

Defines the name of the address-block. This value is defined when you create the address block.

### policy [nameref\(0\)](#) default = default, required

Identifies the name of the policy associated with this address-block.

### segment-name [string](#)

Designates the LAN segment name for this DHCP address block. To group multiple, logical IP subnets on a single, physical network, give each DHCP address block the same segment-name string. The server ignores character case when comparing values.

### selection-tags [string](#)

Lists tag strings that are compared with incoming selection tags in an allocation request. All tags in the request must match a DHCP address block's selection tags for that block to satisfy the request. Separate multiple tags with a comma (do not include commas in tag names).

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

### vpn-id [int](#) default = 0, immutable

Sets the VPN identifier for the VPN that contains this address-block.

## dhcp-address-block-policy

`dhcp-address-block-policy` - Edits the DHCP policy embedded in a `dhcp-address-block`

### Synopsis

```
dhcp-address-block-policy <name> delete
dhcp-address-block-policy <name> set <attribute>=<value>
                               [<attribute>=<value> ...]
dhcp-address-block-policy <name> get <attribute>
dhcp-address-block-policy <name> disable <attribute>
dhcp-address-block-policy <name> enable <attribute>
dhcp-address-block-policy <name> show

dhcp-address-block-policy <name> setLeaseTime<time-val>
dhcp-address-block-policy <name> getLeaseTime

dhcp-address-block-policy <name> setOption <opt-name | id> <value>
                               [-blob | -expression] [-roundrobin]
dhcp-address-block-policy <name> getOption<opt-name | id>
dhcp-address-block-policy <name> unsetOption<opt-name | id>
dhcp-address-block-policy <name> listOptions

dhcp-address-block-policy <name> setVendorOption <opt-name | id>
                               <opt-set-name> <value> [-blob | -expression]
dhcp-address-block-policy <name> getVendorOption <opt-name | id>
                               <opt-set-name>
dhcp-address-block-policy <name> unsetVendorOption <opt-name | id>
                               <opt-set-name>
dhcp-address-block-policy <name> listVendorOptions
```

### Description

The `dhcp-address-block-policy` command lets you configure a DHCP policy embedded in a DHCP address block. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case an address block. An `dhcp-address-block-policy` is created implicitly when you first reference it, and is deleted when the address block is deleted.

See the help file for the policy command for more information.

## Examples

## Status

## See Also

[policy](#), [client-policy](#), [client-class-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the inhibit-all-renews attribute or the inhibit-renews-at-reboot attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

### allow-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

### default-prefix-length [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

### excluded-prefix [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603).

Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified.  
Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64.  
If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

#### **forward-dnsupdate** [nameref\(0\)](#)

Specifies the name of the update configuration that determines which forward zones to include in updates.

#### **forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

#### **giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

#### **grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

#### **inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

#### **inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

#### **lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

#### **limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

#### **longest-prefix-length** [rangeint\(0-128\)](#)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### **max-client-lease-time** [rangetime\(60s-2y\)](#)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1).  
The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

#### **max-client-rebinding-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server.  
The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

#### **max-client-renewal-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client renewal time (T1) that the server is

allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

#### max-leases-per-binding [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

#### offer-timeout [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

#### packet-file-name [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

#### packet-server-name [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

#### packet-siaddr [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

#### permanent-leases [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

#### preferred-lifetime [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

#### reconfigure [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- |            |   |
|------------|---|
| 1 allow    | Allows clients to request reconfiguration support and the server will honor the request (default).  |
| 2 disallow | Allows clients to request reconfiguration support but the server will not honor the clients' request.   |
| 3 require  | Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option. |

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

#### reconfigure-via-relay [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is

available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

#### **reverse-dnsupdate** [nameref\(0\)](#)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### **server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### **shortest-prefix-length** [rangeint\(0-128\)](#)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### **split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### **tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### **unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system\_default\_policy, then the default is 86400 seconds (or 24 hours).

#### **use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### **v4-bootp-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

#### **v4-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### **v6-max-client-preferred-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### **v6-max-client-rebinding-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### **v6-max-client-renewal-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value.

This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### v6-max-client-valid-lifetime [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsoo-allowed-options [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, It overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## dhcp-dns-update

dhcp-dns-update - Configures a DNS Update object for DHCP

### Synopsis

```
dhcp-dns-update <name> create [<attribute>=<value>]
dhcp-dns-update <name> delete
dhcp-dns-update list
dhcp-dns-update listnames
dhcp-dns-update listbrief
dhcp-dns-update <name> show
dhcp-dns-update <name> set <attribute>=<value> [<attribute>=<value> ...]
dhcp-dns-update <name> get <attribute>
dhcp-dns-update <name> unset <attribute>

dhcp-dns-update <name> enable <attribute>
dhcp-dns-update <name> disable <attribute>

dhcp-dns-update < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
dhcp-dns-update < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
dhcp-dns-update <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The dhcp-dns-update command configures dynamic DNS update configurations for DHCP. DHCP policies refer to this information to control the DNS update behavior for each client.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".



## Examples

## Status

## See Also

[policy](#)

## Attributes

### backup-server-addr [ipaddr](#)

Specifies the backup DNS server address that receives DNS updates if the server specified in server-addr is down.

### backup-server-ip6address [ip6addr](#)

Specifies the backup DNS server IPv6 address that receives DNS updates if the server specified in server-ip6address is down.

### backup-server-key [nameref\(0\)](#)

Specifies the TSIG key used to process all dynamic DNS updates for backup-server-addr or backup-server-ip6address.

### dns-client-identity [enumint](#)(txt=1, dhcid=2, transition-to-dhcid=3, regress-to-txt=4)

Specifies the type of resource record (RR) the server uses to identify clients in DNS updates to avoid client naming conflicts.

- 1 - txt The server will use the TXT RR in DHCPv4 DNS updates. This setting is used for backwards compatibility as pre-8.2 versions of CNR only support using TXT RRs for DHCPv4. This setting must be used if any pre-8.2 clusters are involved in doing DNS updates to the zone(s).
- 2 - dhcid The server will use the DHCID RR for DHCPv4 DNS updates. This setting should be used to support dual stack clients and can only be used if all DHCP servers doing DNS updates to the zone(s) for this configuration support and are configured to use the DHCID RR.
- 3 - transition-to-dhcid The server will use the DHCID RR for new entries in DNS and update existing entries to use the DHCID RR on the next DNS update done. This setting is required for a period of time to transitioning existing zones (which used TXT RRs) and this can only be done when all servers doing DNS updates have been upgraded to support use of the DHCID RR (8.2 and later). See below on how to expedite this transition.
- 4 - regress-to-txt The server will use the TXT RR for new entries in DNS and upgrade existing entries to use the TXT RR on the next DNS update done. This setting is provided for use in cases where zones were prematurely transitioned to use the DHCID RR. See below on how to expedite this transition.

For the "transition-to-dhcid" and "regress-to-txt", it is recommend to use the following procedure to expedite the transition to the new RR type as quickly as possible:

1. Set the dns-client-identity attribute to "transition-to-dhcid" (or "regress-to-txt").
2. Note the value of the force-dns-update attribute.
3. Enable the force-dns-update attribute (set it to true), if not already enabled.
4. Reload the server.
5. After a period of time of at least the longest lease time configured in the server (for the leases in the zones being updated), set the dns-client-identity attribute to "dhcid" (or "txt") and restore the force-dns-update attribute to its earlier value if it was changed.

If this attribute value is not set, the DHCP server's dns-client-identity attribute is used.

### dns-host-bytes [rangeint](#)(1-4)

Sets the number of bytes in a lease IP address to use when forming in-addr.arpa names. The server forms names in the in-addr.arpa zone by prepending these bytes of the address (in reverse order) to the reverse zone name. If unset, the server synthesizes an appropriate value based on the scope's subnet size.

### dynamic-dns [enumby](#) default = update-all

Controls whether the DHCP server should attempt to update a DNS server with the name and address information from leases that are granted to requesting clients.

- update-none Do not perform any DNS updates.
- update-all Update both the forward and reverse zones.
- update-fw-only Update only the forward zone (with the A/AAAA).
- update-reverse-only Update only the reverse zone (with the PTR).



**force-dns-updates** [bool](#) default = false

Determines whether the DHCP server retries a dynamic DNS update whenever a client renews its lease, even if DHCP thinks that the update has already been completed successfully.

**forward-zone-name** [dname](#)

Designates the DNS forward zone used to update DNS with forward (A/AAAA) or reverse (PTR) record(s).

**host-name-generator** [expr](#)

Defines an expression that evaluates to the synthesized host name to be used. If undefined or the expression returns "", the configured host name synthesis is used. If the expression returns a null value, no host name will be used.

**max-dns-ttl** [time](#)

Indicates the maximum number of seconds the DHCP server keeps DNS records it acquired through dynamic updates. This value sets a ceiling (or time to live) on how long to keep DNS updates. When the DHCP server adds a DNS record, it uses a TTL of one third the lease time if it is between min-dns-ttl and max-dns-ttl values. If one third of the lease time is greater than max-dns-ttl, the TTL value is set to max-dns-ttl. When this value is unset, the DHCP server max-dns-ttl setting will apply.

**min-dns-ttl** [time](#)

Indicates the minimum number of seconds the DHCP server keeps the DNS records acquired through dynamic updates. This value sets the shortest allowable time (or time to live) to keep DNS updates. When the DHCP server adds a DNS record, it uses a TTL of one third the lease time if it is between min-dns-ttl and max-dns-ttl values. If one third of the lease time is smaller than min-dns-ttl, the TTL value will be set to min-dns-ttl.

**name** [string](#) required,unique

Gives a unique name to the configuration object.

**prefix-delegation-updates** [bool](#) default = disabled

Controls whether to perform DNS updates for delegated prefixes. By default, this feature is disabled for compatibility with earlier releases which does not support this feature. When enabled, the base IPv6 address is used to update DNS (an AAAA and/or PTR RR created based on the dynamic-dns settings). This feature is useful for service providers that need to provide reverse DNS information, such as to third party access providers, to enable use of DNS for looking up information about IPv6 delegated prefixes (such as some information associated with the subscriber in the PTR's data).

**reverse-zone-name** [dname](#)

Designates the DNS reverse (in-addr.arpa) zone that is updated with PTR records. If a reverse-zone-name is configured, DHCP always uses it. Alternately, if DHCP server synthesize-reverse-zone feature is enabled and if a reverse-zone-name is not configured, DHCP uses the lease IP address and the scope dns-host-bytes to automatically generate a reverse zone name.

**reverse-zone-prefix-length** [rangeint](#)(0-124)

Identifies the prefix length of the reverse zone for ip6.arpa updates. The server forms the zone name using this value if configured; otherwise the prefix length is determined from the Prefix. This value must be a multiple of 4 as ip6.arpa zones are on 4 bit (nibble) boundaries. If not a multiple of 4, it is rounded up to the next higher multiple of 4.

**server-addr** [ipaddr](#)

Specifies the DNS server address that receives dynamic DNS updates.

**server-ip6address** [ip6addr](#)

Specifies the DNS server IPv6 address that receives dynamic DNS updates.

**server-key** [nameref](#)(0)

Specifies the TSIG key used to process all dynamic DNS updates for server-addr or server-ip6address.

**synthesize-name** [bool](#) default = disabled

Controls whether the DHCP server automatically creates DNS host names for DHCP clients that do not provide names. The server can synthesize unique names for clients based on the 'synthetic name stem' attribute.

**synthetic-name-stem** [string](#) default = dhcp

Identifies the stem of the default host name to use if clients do not

supply host names.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**update-dns-for-bootp** [bool](#) default = enabled

Controls whether the DNS server receives updates for BOOTP clients. If the server is replying to a BOOTP request, and is offering a lease configured to perform DNS updates, it checks this attribute before beginning the DNS update. This attribute allows an administrator to prevent DNS updates for BOOTP clients, while allowing updates for DHCP clients. If not configured, the server setting is used (which defaults to true).

**use-dns-update-prereqs** [bool](#) default = true

If configured, the update ignores DHCP server level use-dns-update-prereqs. By default (enabled), the DHCP server adds prerequisites to DNS update messages to assure that the current client uses the domain name, before updating the client with the current lease (IP Address). If disabled, the DHCP server assumes that the current client is entitled to the domain name (and hence not add prerequisites to DNS update message) and associates the client's current lease with that domain name. The server only uses this for DHCPv4 updates; DHCPv6 updates always use prerequisites. Note: The DHCP server always adds prerequisites to the DNS update message while adding a new domain name record on behalf of a client acquiring a new lease and removing the domain name record when client release its lease or the lease expires.

**use-gss-tsig** [bool](#) default = false

Indicates whether gss-tsig should be used for updates.

**v4-synthetic-name-generator** [enumint](#)(address=1, client-id=2, hashed-client-id=3) default = address

Controls what the DHCP server appends to the synthetic name stem when synthesizing the fully-qualified domain name for a client. This option is used when client does not supply any hostname in its request.

address:

Identifies the v4 address of client.

client-id:

Client-id or DUID given by DHCPv4 client in its request (Option 61).

hashed-client-id:

The hashed client id. The rightmost 64-bits of the SHA-256 hash over the client id appended with the forward zone name (in DNS wire format) is used to generate a 13-character base 32 encoded string.

**v6-host-name-generator** [expr](#)

Defines an expression that evaluates to the synthesized host name to be used. If undefined or the expression returns "", the configured host name synthesis is used. If the expression returns a null value, no host name will be used.

**v6-synthetic-name-generator** [enumint](#)(duid=1, hashed-duid=2, cablelabs-device-id=3, cablelabs-cm-mac-addr=4) default = hashed-duid

Controls what the DHCP server appends to the synthetic name stem when synthesizing the fully-qualified domain name for a client:

duid

The raw client DUID. This is formatted as a hexadecimal string with a hyphen as separator between each octet.

hashed-duid

The hashed client DUID. The rightmost 64-bits of the SHA-256 hash over the client's DUID appended with the forward zone name (in DNS wire format) is used to generate a 13-character base 32 encoded string. This is the default method and the fallback method if the configured type is not possible (that is, the option needed to generate the selected type does not exist for the client).

cablelabs-device-id

The Cablelabs CL OPTION DEVICE ID option data is used, if available. This is formatted as a hexadecimal string with a hyphen as separator between each octet. This might be used for names generated for DOCSIS 3.0 cable modems.

cablelabs-cm-mac-addr

The Cablelabs CL CM MAC ADDR option data is used, if available. This is formatted as a hexadecimal string with a hyphen as separator between each octet. This might be used for names generated for CPEs behind a customer's DOCSIS 3.0 cable modem (if the customer has multiple CPEs, a name collision and disambiguation will likely result).

Note: Several of these methods may cause privacy concerns if the DNS is accessible from the Internet.

**view-id** [int](#)

Identifies the view associated with the forward and reverse zones used for DNS update.

## dhcp-interface

dhcp-interface - Configures the DHCP server's network interfaces

## Synopsis

```
dhcp-interface <name> create [<attribute>=<value>]
dhcp-interface <name> delete
dhcp-interface list
dhcp-interface listnames
dhcp-interface listbrief
dhcp-interface <name> show
dhcp-interface <name> set <attribute>=<value> [<attribute>=<value> ...]
dhcp-interface <name> get <attribute>
dhcp-interface <name> unset <attribute>

dhcp-interface <name> enable <attribute>
dhcp-interface <name> disable <attribute>
```

## Description

The dhcp-interface command configures network interfaces for use by the Network Registrar DHCP server. If there are no defined interfaces, the server discovers and uses all available interfaces on the system. When this list is present, the server uses only the available interfaces, if any, that match this list.

## Examples

## Status

## See Also

### Attributes

#### address [subnet](#)

Specifies the IP address and subnet mask length of one or more interfaces that the DHCP server should use.

#### ip6address [prefix](#)

Specifies the IPv6 address and prefix length of one or more interfaces that the DHCP server should use.

#### multicast-addresses [nlist\(obj\(0\)\)](#) default = ff02::1:2,ff05::1:3

Enables or disables the specified multicast addresses on DHCP interfaces. The default multicast addresses are ff02::1:2 and ff05::1:3. DHCPv6 requires address ff02::1:2, if any DHCPv6 clients are directly connected to the link associated with the interface. The address ff05::1:3 is the default multicast addresses used by relay agents when relaying DHCPv6 requests.

#### name [string](#) required,unique

Specifies the user-assigned name of the DHCP server interface.

## dhcp-listener

dhcp-listener - Configures a TCP dhcp-listener for DHCP.

## Synopsis

```
dhcp-listener <name> create <address>
                                     [<attribute>=<value> ...]
dhcp-listener <name> delete
dhcp-listener list
dhcp-listener listnames
dhcp-listener listbrief
dhcp-listener <name> show
dhcp-listener <name> set <attribute>=<value>
                                     [<attribute>=<value> ...]
dhcp-listener <name> get <attribute>
dhcp-listener <name> unset <attribute>
```

```
dhcp-listener <name> enable <attribute>
dhcp-listener <name> disable <attribute>
```

## Description

The **dhcp-listener** command configures objects to enable active and bulk leasequery to the DHCP server over TCP connections. To enable active and bulk leasequery, one of these objects must be created. Multiple objects can be used to accept connections on different ports.

## Examples

## Status

## See Also

### Attributes

address [ipaddr](#)

Specifies the address to which the service is bound. To accept connections to any valid local address, specify 0.0.0.0. If both address and ip6address are unset, the IPv4 address 0.0.0.0 will be used. However, both cannot be set. Create separate objects, one for IPv4 and another for IPv6 to use the same port for both.

enable [bool](#) default = true

Specifies whether this service is enabled. If disabled, the DHCP server will ignore this listener configuration.

ip6address [ip6addr](#)

Specifies the address to which the service is bound. To accept connections to any valid local address, specify :: (the all 0's address). If both address and ip6address are unset, the IPv4 address 0.0.0.0 will be used. However, both cannot be set. Create separate objects, one for IPv4 and another for IPv6 to use the same port for both.

leasequery-backlog-time [rangetime](#)(30s-10m) default = 2m

Specifies the number of seconds of active leasequery updates that the DHCP server will hold in memory when a connection is blocked. If multiple listener objects are configured, the DHCP server uses the longest time for this attribute for all its connections.

leasequery-send-all [bool](#) default = false

Specifies whether to send a message to active leasequery clients for every write to the DHCP lease state database. The alternative (and the default) is to optimize the traffic over active leasequery connections and to send only updates which are necessary to maintain accurate state in the active leasequery client. The optimizations are largely involved with failover updates, in an attempt to avoid redundant active leasequery messages. However, this optimization is only valid if the active leasequery client connects to both failover servers. Should you have an active leasequery client which only ever connects to one failover partner, this optimization will prevent important data from reaching such a client and you should enable this attribute to prevent the optimization. Configuring any listener with this attribute enabled will cause all active leasequery connections to send a message for every write to the DHCP lease state database.

max-connections [int](#) default = 10

Specifies the maximum number of simultaneous connections allowed for the service.

name [string](#) required,unique

Specifies the name of service. This name has no significance.

port [short](#)

Specifies the port number on which the DHCP server listens for TCP connections. The default port is the server-port for DHCPv4 and v6-server-port for DHCPv6.

# dhcp-maintenance-window

dhcp-maintenance-window - configures a DHCP maintenance window

## Synopsis

```
dhcp-maintenance-window list
dhcp-maintenance-window listnames
dhcp-maintenance-window listbrief
dhcp-maintenance-window <name> create [<attribute>=<value> ...]
dhcp-maintenance-window <name> delete
dhcp-maintenance-window <name> set <attribute>=<value>
    [<attribute>=<value> ...]
dhcp-maintenance-window <name> get <attribute>
dhcp-maintenance-window <name> unset <attribute>
dhcp-maintenance-window <name> enable <attribute>
dhcp-maintenance-window <name> disable <attribute>
dhcp-maintenance-window <name> [show]

dhcp-maintenance-window clearMaintenance [ dhcpv4 | dhcpv6 ]

dhcp-maintenance-window addScopes <name1>[,<name2>[,...]]
dhcp-maintenance-window removeScopes <name1>[,<name2>[,...]]
dhcp-maintenance-window listScopes

dhcp-maintenance-window addPrefixes <name1>[,<name2>[,...]]
dhcp-maintenance-window removePrefixes <name1>[,<name2>[,...]]
dhcp-maintenance-window listPrefixes

dhcp-maintenance-window addLinks <name1>[,<name2>[,...]]
dhcp-maintenance-window removeLinks <name1>[,<name2>[,...]]
dhcp-maintenance-window listLinks
```

## Description

The dhcp-maintenance-window command configures a DHCP maintenance window. There can only be a single DHCP maintenance window.

The dhcp-maintenance-window defines a period of time when the DHCP server is expected to be out of service. The DHCP server will adjust renewal times from existing DHCP clients to avoid this window, and to renew in the period immediately after the window is over in order to pick up potential new information.

If the all-scopes is not set, only scopes (and all scopes for the network the scope is on) that have the maintenance attribute set to true will be subject to the maintenance window; clients on other scopes will not participate.

If the all-prefixes-links is not set, only prefixes (and all prefixes for that link) or the links that have the maintenance attribute set to true will be subject to the maintenance window; clients on other prefixes and links will not participate.

The clearMaintenance command can be used to unset (clear) the maintenance attribute on all scopes, all prefixes/links, or all scopes and prefixes/links.

The addScopes, addPrefixes, and addLinks command can be used to set the maintenance attribute for scope, prefix, or link objects.

The removeScopes, removePrefixes, and removeLinks command can be used to unset the maintenance attribute for scope, prefix, or link objects.

The listScopes, listPrefixes, and listLinks commands can be used to see which objects have their maintenance attribute enabled. Note that these commands ignore the all-scopes or all-prefixes-links setting and only include the objects that explicitly have their maintenance attribute enabled.

Note: Deleting the maintenance window object unsets the maintenance attribute on all scopes, all prefixes, and all links.

## Examples

## Status

## See Also

[scope](#), [prefix](#), [link](#)

## Attributes

**all-prefixes-links** [bool](#) default = disabled

If this attribute is enabled, all prefixes and links in the configuration will participate in the current maintenance window. If not enabled, only links that have their maintenance attribute set to enabled or a prefix in the link has its maintenance attribute set to enabled, will participate.

**all-scopes** [bool](#) default = disabled

If this attribute is enabled, all scopes in the configuration will participate in the current maintenance window. If not enabled, only scopes that have their maintenance attribute set to enabled will participate.

**end** [date](#)

The date and time at which the maintenance window ends. The end date must be after the start date and within 24 hours after the start (the maintenance window must be less than 24 hours).

**name** [string](#) required,unique

The name of this DHCP maintenance window object.

**refresh-period** [time](#)

The period after the end of the maintenance window when all the affected clients should contact the server to pick up any newly configured information. At the end of the refresh period, all existing clients should have contacted the server, or their leases should have expired. This period must be long enough so that having all affected DHCP clients renewing during 1/2 of this period will not place an unacceptable load on the DHCP server. It should be several hours at the very least.

**start** [date](#)

The date and time at which the maintenance window starts. The start date must be far enough in the future from when the maintenance window is configured to allow the server to interact with all affected DHCP clients before the start date is reached. Typically, this means that the time between configuring the maintenance window and the start date must be longer than the longest lease time of any affected DHCP client.

## dhcp-subnet

`dhcp-subnet` - Describes a contiguous range of IP address space which the DHCP server has allocated to a client.

## Synopsis

```
dhcp-subnet <dhcp-subnet-number> [show]  
dhcp-subnet <subnet-number> get <attribute>  
dhcp-subnet <subnet-number> force-available
```

## Description

The `dhcp-subnet` commands manipulate subnet objects that the DHCP server has leased to its clients. When the DHCP server receives a request for a subnet, it creates the subnet by subdividing its available address-blocks, and allocating the subnet to the client.

The `dhcp-subnet` commands apply only to subnet objects that have been allocated to clients by the DHCP server.

## Examples

```
dhcp-subnet 10.10.1.0/24 show
```

## Status

### See Also

[dhcp-address-block](#)

### Attributes

**address** [subnet](#) immutable

Displays the subnet address, including the mask.

**all-vpns** [bool](#)

Provides failover configuration information for this subnet. If true, this subnet is configured to use the same failover configuration for all VPNS. Read only.

**client-domain-name** [string](#)

Displays the domain name the client specified in its messages (if any).

**client-flags** [flags](#)(client-valid=1, client-id-created-from-mac-address=2)

Displays either client-valid or client-id-created-from-mac-address (the client ID was created for internal use from the client MAC address).

**client-host-name** [string](#)

Displays the host name that the client specified (if any).

**client-id** [blob](#)

Displays the client-id of the subnet's client.

**client-last-transaction-time** [date](#)

Displays the time when the client last contacted the DHCP server.

**client-mac-addr** [macaddr](#)

Displays the MAC address which the client presented to the DHCP server.

**expiration** [date](#)

Displays the expiration time of the subnet binding.

**high-water** [int](#)

Displays the highest utilization level recorded since the last time that statistics were retrieved.

**in-use-addresses** [int](#)

Displays the number of addresses currently being used by hosts.

**last-transaction-time** [date](#)

Displays the time at which the client last communicated with the server about this subnet. Read only.

**relay-agent-option** [blob](#)

Displays the contents of the relay agent information option from the most recent client interaction. Read only.

**selection-tags** [string](#)

Displays the selection-tag string that the client presented when it last leased or renewed the subnet binding. Read only.

**state** [enumint](#)(none=0, available=1, other-available=2, offered=3, leased=4, expired=5, released=6, unavailable=7, pending-available=8)

Displays the state of the subnet.

**tenant-id** [short](#) default = 0

Identifies the tenant owner of this object.

**unusable-addresses** [int](#)

The number of addresses marked unusable.

vpn-id [int](#) default = 0

Identifies the VPN that contains this subnet.

## dns

dns - Configures and controls the DNS server

### Synopsis

```
dns disable <attribute>
dns enable <attribute>

dns get <attribute>
dns set <attribute>=<value> [attribute]=<value ...]
dns unset <attribute>
dns show

dns findRR -name <fqdn>|<addr>
dns findRR [-namePrefix <namePrefix>] [-rrTypes <rrTypeList>]
           [-protected | -unprotected] [-zoneType forward | reverse
           | primary | secondary | ALL]

dns rebuildRR-Indexes

dns forceXfer secondary

dns scavenge

dns serverLogs show
dns serverLogs nlogs=<nlogs> logsize=<logsize>

dns getStats [ <performance [,] query [,] update [,] errors [,] security [,]
              maxcounters [,] ha [,] ipv6 [,] [,] cache [,] datastore [,]
              top-names [,] dns-hhc | all> [total | sample] ]

dns resetStats

dns getUtilization

dns getZoneCount [forward | reverse | primary | secondary | ALL]
dns getRRCount [zone <name> | forward | reverse | primary | secondary | ALL]

dns setPartnerDown

dns rollover-ksk [tenant-id=<value>] [next-key=<keyname> | key-group=<value>]
```

### Description

The dns command lets you configure the DNS server in the cluster.

```
dns findRR -name <fqdn>|<addr>
dns findRR [-namePrefix <namePrefix>] [-rrTypes <rrTypeList>]
           [-protected | -unprotected] [-zoneType forward | reverse
           | primary | secondary | ALL]

Use the findRR commands to display the resource records for a
specific domain name; or to display those matching a name prefix,
a list of resource record types--whether protected or unprotected--and
certain zone types.
```

**dns rebuildRR-Indexes**  
The rebuildRR-Indexes command rebuilds the resource record indexes.

**dns forceXfer secondary**  
The forceXfer command forces full zone transfers for every zone whose type matches the type (primary or secondary) specified in the command, regardless of the SOA serial numbers, to synchronize DNS data store. If a normal zone transfer is already in progress, the forceXfer command schedules a full zone transfer for that zone immediately after the normal zone transfer finishes.

Note: The option for primary is not yet available.

**dns scavenge**  
The scavenge command causes scavenging to occur on all primary zones that have scavenge enabled.

**dns serverLogs show**



### **dns serverLogs nlogs=<nlogs> logsize=<logsize>**

The serverLogs show command displays the number of log files and the maximum size for each file.

The serverLogs command allows setting the two server logging parameters, nlogs and logsize. Either or both may be specified in the command, and changes will only occur to the one(s) specified. When setting logsize, a suffix of K or M indicates units of thousands or millions.

```
dns serverLogs nlogs=6 logsize=500K
dns serverLogs logsize=5M
```

Note: For these changes to take effect you must save the changes and restart the server Agent.

### **dns getStats [<statistics-categories> [total | sample ]**

The getStats command displays the requested DNS server statistics, either since the last reload or for the last sample period. You can supply one or more specific categories of statistics counters (performance, query, update, errors, security, maxcounters, ha, ipv6, cache, datastore, top-names, dns-hhc), or the keyword all to retrieve all supported categories.

### **dns resetStats**

The resetStats commands returns the DNS activity counters (statistics) to zero.

### **dns getUtilization**

The getUtilization command can be used to get the count of total number of A and AAAA record for all zones.

### **dns getZoneCount [forward | reverse | primary | secondary | ALL]**

### **dns getRRCount [zone <name> | forward | reverse | primary | secondary | ALL]**

The getZoneCount and getRRCount commands display the number of zones or resource records for the requested zones. By default, all published zones are reported.

### **dns setPartnerDown**

The setPartnerDown command notifies the DNS server that its High Availability DNS partner server is down. This command is only allowed once the DNS server is in Communications Interrupted state. Once in Partner Down state, the DNS server no longer persists changes that need to be merged with its DNS HA partner and must send FULL zone synchronization when the partner comes back online.

### **dns rollover-ksk**

The rollover-ksk command is used to start the Key Signing Key rollover process. The tenant-id must be specified to rollover a tenant Key Signing Key. **next-key=<keyname>** can be used to specify the new Key Signing Key. Otherwise, a new Key Signing Key will be generated internally.

## Examples

## Status

## See Also

[server](#)

## Attributes

**acl-blocklist** [amelist](#) default = none

Blocks requests from clients listed in this access control list. This list can contain hosts, network addresses and/or other ACLs. Request from clients matching this acl will be dropped.

**activity-counter-log-settings** [flags](#) (performance=3, query=4, errors=5, security=6, maxcounters=7, ha=8, ipv6=9, db=10, system=11, host-health-check=13, top-names=14, cache=15, update=16) default = performance,query,errors,maxcounters,system,update

Controls what activity counters a DNS server uses for logging. The possible flags are:

host-health-check	log DNS Host Health Check counters.
cache	log query cache related counters.
db	log database counters.
errors	log error related counters.

ha	log HA related counters.
ipv6	log IPv6 related counters.
maxcounters	log maxcounters related counters.
performance	log performance related counters.
query	log query related counters.
security	log security related counters.
system	log system related counters.
top-names	log the top names queried and hit count.
update	log DNS Update related counters.

**activity-summary-interval** [time](#) default = 60s

Sets the seconds between DNS activity summary log messages, if the activity-summary flag is enabled in server-log-settings.

**activity-summary-type** [flags](#)(sample=1, total=2) default = sample

Determines whether the DNS server logs sample and/or total statistics when it logs activity-summary information. Note, activity-summary must be specified in the log-settings in order for this setting to take effect.

**allow-any-query-acl** [amelist](#) default = none

Allows ANY query requests from clients listed in this access control list. This list can contain hosts, network addresses and/or other ACLs. Request from clients matching this acl will be allowed ANY query.

**contrack-usage-warning** [percent](#) default = 50%

Specifies the percentage of the firewall connection tracking usage that will trigger a warning. Warnings are logged at the activity-summary-interval whenever the number of connections is at or above the specified percentage. Setting the value to 0 disables the warnings. Note: This attribute is only applicable to Linux systems.

It is highly recommended that connection tracking not be used for DNS related activity as this can block legitimate traffic (incoming or outgoing) and result in resolution failures for clients or other DNS servers.

It is also strongly recommended that the firewall be completely disabled when using the DNS server as this can cause significant performance impact to DNS services especially for high query loads.

**drop-recursive-queries** [bool](#) default = disabled

Controls whether the DNS server accept or drops queries which have rd flag on. When the setting is enabled, recursive queries will be dropped by the server. The default value will be disabled, meaning that no recursive queries will be dropped.

**edns-max-payload** [rangeint](#)(512-4096) default = 1232

Specifies the sender's maximum UDP payload size, which is defined as the number of octets of the largest UDP packet that can be handled by a requestor (See RFC 6891).

**gss-tsig-config** [nameref](#)(0)

Identifies the gss-tsig configuration object to be used by DNS server.

**gss-tsig-processing** [flags](#)(ddns=1, query=2, none=3) default = none

Indicates the gss-tsig security mode for DNS transactions. If both gss-tsig-processing and tsig-processing are enabled, gss-tsig security mode will be disabled. Default is none (disabled).

**hhc-interval** [rangetime](#)(60s-24h) default = 30m

Specifies the time interval (in seconds) to check RR sets for reachability.

**hhc-max-init-ttl** [rangetime](#)(60s-30m) default = 60s

Specifies the maximum initial TTL(in seconds) to send in query reply when Host Health Check RR is queried for the first time. Note that if the RR set has a TTL less than hhc-max-init-ttl, the RR set's TTL will be used in the response.

**hhc-max-ttl** [rangetime](#)(60s-24h) default = 30m

Specifies the maximum TTL (in seconds) to send in query reply when RR health status is up. Note that if the RR set has a TTL less than hhc-max-ttl, the RR set's TTL will be used in the response.

**host-health-check** [bool](#) default = disabled

Enables or disables DNS Host Health Check in the DNS server. When Host Health Check is enabled, DNS server sends hhc-max-ttl as ttl in query reply for active RRs. When DNSSEC is enabled, dns server will

add RRs which are not active at the end of RR list  
in query reply.  
When DNSSEC is not enabled, dns server will  
not add RRs which are not active in RR list  
in query reply.

**hybrid-adns-addr** [nlist](#)(obj(0)) default = 127.0.0.1

Specifies a list of one or more IP addresses to use for hybrid mode communication. Note that these addresses should match one or more of the Authoritative DNS server's configured interfaces.  
Note: If using addresses other than the loopback address (127.0.0.1), it may be necessary to also configure these interfaces in the Caching DNS Server for outbound traffic.

**hybrid-mode** [bool](#) default = disabled

Enables DNS Hybrid mode which allows Caching and Authoritative DNS servers to run on the same physical or virtual machine. Note that hybrid-mode requires that separate network interfaces be configured on both the Caching and Authoritative DNS servers.

**hybrid-override** [enumint](#)(zones-override-exceptions=1, exceptions-override-zones=2) default = zones-override-exceptions

Specifies whether the Caching DNS server configures hybrid-mode zones or exceptions when hybrid-mode is enabled and there are CDNS exceptions that match DNS zones.

**hybrid-use-adns-addr** [bool](#) default = false

When enabled, the Caching DNS server will setup hybrid exceptions to forward to the Authoritative DNS server via hybrid-adns-addr. The hybrid-adns-addr attribute defaults to the loopback address (127.0.0.1) which is the recommended interface for hybrid DNS communication.  
If disabled, the Caching DNS server will use all of the Authoritative DNS server's configured network interfaces.

**max-dns-packets** [int](#) default = 5000

Specifies the maximum number of packets that dns server will handle concurrently. DNS server will drop inbound packets if this limit is reached.

**mem-cache-size** [int](#) default = 500000

Indicates the maximum size, in kilobytes, of the in-memory resource record cache.

**minimal-responses** [bool](#) default = false

Controls whether the DNS server omits records from the authority and data sections of query responses when these records are not required. Enabling this attribute may improve query performance, reduce query response size and avoid TCP fallback.

**minimize-ttls** [bool](#) default = false

Ensures that queries responses have RRs with the same (minimized) TTL.

**notify** [enumbyt](#) default = notify-all

Controls how the DNS server sends NOTIFY packets for zones that have changed.  
You must also set these attributes or accept their defaults: notify-min-interval, notify-rcv-interval, notify-send-stagger, and notify-wait.  
Possible settings are:  
disabled  
Notifications will be turned off.  
notify-all  
Notifications will be sent to all NS and notify-list servers.  
notify-ns  
Notifications will only be sent to NS servers.  
notify-list  
Notifications will only be sent to the notify-list servers.

**notify-min-interval** [rangetime](#)(0-68y5w3h14m7s) default = 2s

With the notify attribute enabled, sets the minimum interval required before sending notification to a particular server of consecutive changes on the same zone. Default is 2s.

**notify-rcv-interval** [rangetime](#)(0-68y5w3h14m7s) default = 5s

With the notify attribute enabled for secondary zones, sets the minimum amount of time between complete processing of one notification (serial number testing and/or zone transfer), and the start of processing of another notification. , default 5s.

**notify-send-stagger** [rangetime](#)(0-68y5w3h14m7s) default = 1s

With the notify attribute enabled, sets the interval to use for staggering notifications to multiple servers about a zone change. Default is 1s.

**notify-source-address** [ipaddr](#)

Specifies the source IPv4 address that the DNS server uses to send notify requests to other servers. If this attribute is unset, the server uses the best local IPv4 address based on the destination address.

#### **notify-source-ip6address** [ip6addr](#)

Specifies the source IPv6 address that the DNS server uses to send notify requests to other servers. If this attribute is unset, the server uses the best local IPv6 address based on the destination address.

#### **notify-source-port** [rangeint](#)(0-65535) default = 0

Specifies the port number that the DNS server uses to send notify requests to other servers. A value of 0 (default) indicates that DNS should choose a random port. If the value is set to be the same as the query-source-port, DNS will log a warning and choose a random port.

#### **notify-wait** [rangetime](#)(0-68y5w3h14m7s) default = 5s

With the notify attribute enabled, and after an initial zone change, sets the period of time for the DNS server to wait before it sends change notification to other name servers. Default is 5s. This property allows you to accumulate multiple changes, and thus limit the number of times the serial number advances.

#### **packet-log-settings** [flags](#)(all-in=1, all-out=2, query-in=3, query-out=4, update-in=5, update-out=6, notify-in=7, notify-out=8, xfr-in=9, xfr-out=10, ha-in=11, ha-out=12, ha-heartbeat-in=13, ha-heartbeat-out=14, ha-frameack-in=15, ha-frameack-out=16) default = all-in,all-out

Determines the type of DNS messages to log if packet logging has been enabled. Packet logging can be enabled by configuring the packet-logging attribute.

**all-in**  
This setting enables logging of all incoming packets.  
NOTE: This is equivalent to enabling all the -in settings.

**all-out**  
This setting enables logging of all outgoing packets.  
NOTE: This is equivalent to enabling all the -out settings.

**ha-in, ha-out**  
These settings enable logging of HA DNS messages except for HA heartbeat and frame ACK messages which are controlled by the ha-heartbeat-in, ha-heartbeat-out and ha-frameack-in, ha-frameack-out settings, respectively.

**ha-heartbeat-in, ha-heartbeat-out**  
These settings enable logging of HA DNS heartbeat messages.

**ha-frameack-in, ha-frameack-out**  
These settings enable logging of HA DNS frame ACK messages.

**notify-in, notify-out**  
These settings enable logging of DNS NOTIFY messages.

**query-in, query-out**  
These settings enable logging of DNS QUERY messages.

**update-in, update-out**  
These settings enable logging of DNS UPDATE messages.

**xfr-in, xfr-out**  
These settings enable logging of DNS IXFR and AXFR messages.

#### **packet-logging** [enumint](#)(disabled=0, summary=1, detail=2) default = disabled

Determines the type packet logging that is logged to the DNS logs. The type of DNS packets logged can be controlled with the packet-log-settings attribute.

**disabled**  
This settings disables logging of DNS packets.

**summary**  
This setting enables one line summary logging of DNS packets.

**detail**  
This setting enables detailed packet tracing of DNS packets.  
NOTE: This setting may significantly increase the amount of information that is logged and should only be used on a temporary basis for debugging purposes.

Note that while packet logging can be helpful for debugging and troubleshooting, it does have an impact on DNS server performance. Therefore, Cisco does not recommend leaving packet logging enabled in production environments.

#### **packet-logging-file** [enumint](#)(none=0, dns=1, packet=2, all=3) default = dns

Determines the destination log of packet log messages when packet logging is enabled.

**dns**  
Packet logging messages are logged to the standard DNS log file (name\_dns\_1\_log\*).

**packet**  
Packet logging messages are logged to a separate DNS packet log file (dns\_packet\_log\*).

**port rangeint(1-65535) default = 53**

Specifies the UDP and TCP port number that the DNS server uses to listen for DNS messages.

**query-types-unwanted nlist(obj(0))**

Specifies the DNS query RR types that the server is not allowed to process. DNS queries with types specified in this list will be dropped. By default all types are allowed. The types can be specified as a string (i.e. A) or the keyword "TYPE" followed by a number (i.e. TYPE1). Query requests containing unwanted types may signal a DoS attack. Therefore when queries are dropped based on type, the server will increment the security-events-dos statistic. The queries-dropped and queries-dropped-unwanted-type statistics will also be incremented. Changes to this attribute take place without requiring a DNS server reload.

**relax-delegation-rules bool default = enabled**

Allows the server to return RRs that are at or below a subzone delegation if the records are configured in an authoritative zone.

**remote-port-num rangeint(1-65535) default = 53**

Specifies the UDP and TCP port number the DNS server uses to send queries to other servers. Default is port 53.

**restrict-query-acl amelist default = any**

Specifies a global access control list (ACL) used to limit device queries that a DNS server must honor. You can restrict query clients based on host IP address, network address, TSIG keys, and access control lists. The default is to allow any client to perform a query.

**restrict-xfer bool default = true**

Limits sending zone transfers to a specific set of hosts. If you restrict zone transfers, use the restrict-xfer-acl attribute to specify the access control list that defines which servers can perform zone transfers.

**restrict-xfer-acl amelist default = none**

Specifies the default access control list, designating who can receive zone transfers, when zone transfers are restricted.

**round-robin bool default = enabled**

Specifies whether you want round-robin cycling of equivalent records in responses to queries. Equivalent records are records of the same name and type. Since clients often only look at the first record of a set, enabling this feature can help balance loads and keep clients from forever trying to talk to an out-of-service host.

**save-negative-cache-entries bool default = enabled**

Controls in-memory caching of negative query (i.e. NXDOMAIN) results.

**scvg-ignore-restart-interval rangeint(2h-24h) default = 2h**

Ensures that the server does not reset the scavenging time with every server restart. Within this interval, Network Registrar ignores the time between when a server went down and its restart. This interval is normally short. The value can range from two hours to one day. With any time longer than that set, Network Registrar recalculates the scavenging period to allow for record updates that cannot take place while the server is stopped. You can also set this attribute on a zone, and the value set on the zone overrides the server setting. Default is 2h.

**scvg-interval rangeint(60m-1y) default = 1w**

Sets the seconds that DNS waits before removing (scavenging) out-of-date address (A) records.

**scvg-refresh-interval rangeint(2h-1y) default = 2w**

When scavenging is enabled, specifies the expiration time for unprotected RRs. The expiration time is calculated based on the time the RR was last modified. Once an RR has expired, it becomes a candidate for scavenging and may be deleted by the scavenging process. Updates to the RR (i.e. dynamic updates, prereq only updates, etc.) will reset the record's modification time and thus recalculate the scavenging expiration time. NOTE: RRs in a protected name set are not scavenged.

**security-event-log-settings flags(configuration=1, packet-inspection=3, rate-limit=4) default = configuration,packet-inspection,rate-limit**

Specifies the DNS security events that should be logged. When the DNS server detects a security event and the related security event log setting is enabled, a log message will be written to the dns security log file. In order for this setting to take effect, the security-event-logging must be enabled. Note that

security-event-logging and security-event-log-settings configuration changes take effect immediately without requiring a DNS server reload.

configuration  
A security event log message will be generated based on DNS server configuration settings (i.e. ACL failures).

packet-inspection  
A security event log message will be generated based on DNS server detecting issues in the request packet. These issues may be detected by basic packet inspection (i.e. packet-inspection setting) or during packet processing. Excessive malformed packets may indicate a DoS attack.

rate-limit  
A security event log message will be generated if the DNS server reaches its limit for concurrent packets (i.e. max-dns-packets). Excessive DNS traffic may indicate an amplification attack.

#### **security-event-logging** [enumint](#)(disabled=0, enabled=1) default = disabled

Enables DNS security event logging based on settings configured in security-event-log-settings. Note that security-event-logging and security-event-log-settings configuration changes take effect immediately without requiring a DNS server reload. Security event log messages are written to the dns\_security\_log file.

#### **security-event-max-qname-size** [rangeint](#)(0-255) default = 0

Specifies the maximum size of a query name (QNAME) allowed. If a longer name is detected, the server will trigger a packet inspection DNS security event for the DNS tunneling category and the query will be refused. A setting of 0 (default) disables query name length checking.

#### **server-log-settings** [flags](#)(config=1, config-detail=2, activity-summary=3, server-operations=4, query=5, update=6, notify=7, xfr-in=8, xfr-out=9, ha=10, scp=11, scavenger=12, db=13, tsig=14, dnssec=16, host-health-check=17) default = config,activity-summary,server-operations,update,xfr-in,xfr-out,ha,scp,scavenger

Determines which events to log in the DNS log files. Default flags are activity-summary, config, update, xfr-in, xfr-out, scp, scavenger, server-operations and ha.

Logging additional detail about events can help analyze a problem. However, leaving detailed logging enabled for a long period can fill up the log files.

The possible settings are:

host-health-check  
This setting enables logging associated with DNS Host Health Check.

activity-summary  
This setting enables logging of DNS statistic messages at the interval specified by activity-summary-interval. The type of statistics logged can be controlled with activity-counter-log-settings and activity-summary-type.

config  
This setting enables logging of DNS server configuration and de-initialization messages.

config-detail  
This setting enables logging of detailed configuration messages (i.e. detailed zone configuration logging).

db  
This setting enables logging of database processing messages. Enabling this flag provides insight into various events in the server's embedded databases.

dnssec  
This setting enables log messages associated with DNSSEC processing.

ha  
This setting enables logging of HA DNS messages.

notify  
This setting enables logging of messages associated with NOTIFY processing.

query  
This setting enabled logging of messages associated with QUERY processing.

scavenger  
This setting enables logging of DNS scavenging messages.

scp  
This setting enables logging associated with SCP messages handling.

server-operations  
This setting enables logging of general server events, such as those pertaining to sockets and interfaces.

tsig  
This setting enables logging of events associated Transaction Signature (TSIG).

update  
This setting enables logging of DNS Update message processing.

xfr-in  
This setting enables logging of inbound full and incremental zone transfers.

xfr-out  
This setting enables logging of outbound full and incremental zone transfers.

**service-key** [string](#)

Defines the file name which contains the private key to be used by DNS for TLS sessions. The file must be in the DNS data directory under `tls` (i.e. `/dns/tls`) and must not be encrypted with a passcode.

**service-pem** [string](#)

Defines the pem file name which contains the public key certificate to be used by DNS for TLS sessions. The file must be in the DNS data directory under `tls` (i.e. `/dns/tls`). Note that if using managed DNS certificates, this attribute will be ignored and should be left unset.

**simulate-zone-top-dynupdate** [bool](#) default = disabled

Enables compatibility with a Windows 2000 Domain Controller. When processing a dynamic update packet, which attempts to add or remove A records from the name of a zone, DNS responds as if the update succeeded, rather than responding with a refusal, as would normally occur due to the protected/unprotected name conflict. No update to the records at the zone name will actually occur, although the response indicates that it has.

**tls** [bool](#) default = disabled

Enables or disables TLS support for DNS. Before enabling TLS, the private key files must be placed in the DNS data directory under `dns/tls` and the `service-key` attribute set. If using managed DNS certificates, the certificate settings will be automatically set. Otherwise, the public certificate file must be placed in the DNS data directory under `dns/tls` and the `service-pem` attribute set. Enabling or disabling TLS service requires a CPNR service restart for the change to take effect.

**tls-port** [rangeint](#)(1-65535) default = 853

The port number on which to provide TCP TLS service. The DNS server will not serve non-TLS queries on this port.

**top-names** [bool](#) default = enabled

Specifies whether to collect top names data. When enabled, a snapshot of the cache hits for the top names that are queried is collected for each interval set by the `top-names-max-age` value. The list of top names that is reported with activity summary statistics is the most current snapshot.

**top-names-max-age** [rangetime](#)(10s-5m) default = 60s

Specifies the maximum age (based on last access time) of a queried name allowed in the list of top names.

**top-names-max-count** [rangeint](#)(5-1000) default = 10

Specifies the maximum number of entries in the list of top names queried. This limit is applied to the lists of top names that are logged or returned as part of activity summary.

**transfer-source-address** [ipaddr](#)

Specifies the source IPv4 address that the DNS server uses to send transfer and SOA requests to other servers. If this attribute is unset, the server uses the best local IPv4 address based on the destination address.

**transfer-source-ip6address** [ip6addr](#)

Specifies the source IPv6 address that the DNS server uses to send transfer and SOA requests to other servers. If this attribute is unset, the server uses the best local IPv6 address based on the destination address.

**transfer-source-port** [rangeint](#)(0-65535) default = 0

Specifies the port number that the DNS server uses to send transfer and SOA requests to other servers. A value of 0 (default) indicates that DNS should choose a random port. If the value is set to be the same as the `query-source-port`, DNS will log a warning and choose a random port.

**traps-enabled** [flags](#)(all=1, server-start=2, server-stop=3, ha-dns-partner-down=4, ha-dns-partner-up=5, ha-dns-config-error=6, primary-not-responding=7, primary-responding=8, secondary-zone-expired=9)

Defines the traps that this server is configured to send.

- 1 all  
Sends notifications for all server events.
- 2 server-start  
Sends notifications whenever the server is started or reinitialized.
- 3 server-stop  
Sends notifications whenever the server is stopped.

- 4 `ha-dns-partner-down`  
Sends notifications whenever the HA DNS partner goes down.
- 5 `ha-dns-partner-up`  
Sends notifications whenever the HA DNS partner becomes available again after going down.
- 6 `ha-dns-config-error`  
Sends notifications when a configuration mismatch between HA DNS partners occurs.
- 7 `primary-not-responding`  
Sends notifications when primary servers stop responding.
- 8 `primary-responding`  
Sends notifications when primary servers start responding again.
- 9 `secondary-zone-expired`  
Sends notifications when a secondary server can no longer claim authority for zone data when responding to queries during a zone transfer.

**tsig-processing** [flags](#)(ddns=1, query=2, none=3, notify=4) default = ddns,query

Enables you to turn on and off tsig processing for DNS transactions. Default is enabled on ddns and query requests.

**update-acl** [amelist](#) default = none

Provides server-level control of which devices can access and update the DNS server. If the access control list is set at the zone level, Network Registrar overrides the server-level setting.

**update-relax-zone-name** [bool](#) default = disabled

Enables DNS updates to specify any zone name in the authoritative zone rather than the exact zone name; thus, relaxing the RFC 2136 restriction on the zone name record in dynamic updates. This attribute allows updates to specify a zone name which is any name within an authoritative zone rather than exactly the name of a zone.

**xfer-ignore-serial-decrease** [int](#) default = 3

Specifies the number of times a secondary will ignore notify and/or incremental zone transfers with a lower serial number. The value is calculated on a per zone basis and a full zone transfer will be requested once we have surpassed the value.

**xfer-rr-meta-data** [bool](#) default = enabled

Controls whether or not zone transfer requests also include a request for RR meta data. RR meta data may include weight, order and protection state which are not part of standard DNS RR data.

## dns-enum-config

`dns-enum-config` - Configures DNS ENUM defaults in the DNS authoritative servers.

### Synopsis

```

dns-enum-config show
dns-enum-config get <attribute>
dns-enum-config set <attribute>=<value>
dns-enum-config unset <attribute>

dns-enum-config addService <type> <subtype> <URI>
  [<order> [preference]]
dns-enum-config removeService <type> <subtype> <URI>
dns-enum-config listServices

```

### Description

This command is used to configure DNS ENUM defaults for the DNS authoritative servers.

### Examples

```

nrcmd> dns-enum-config set zone-template=zt
nrcmd> dns-enum-config get zone-template
nrcmd> dns-enum-config show

```

### Status



## See Also

### Attributes

**default-services** [nlist\(obj\(0\)\)](#)

Specifies a default URI for the listed service types.  
Use the +number keyword in the URI to specify where the number should be inserted when the service is added to an DnsEnumEntrySet.

**number-prefix** [string](#)

Specifies the E.164 number prefix for local domains.  
The prefix will be prepended to numbers that do not specify the beginning '+' character.

**top-level-domain** [dname](#) default = e164enum.net.

Specifies the top-level domain to use when creating e164 zones.  
A domain name ending in .arpa will be created as a reverse zone.  
Any other domain name will be created as a forward zone.

**zone-template** [nameref\(0\)](#)

Specifies the default zone template to use when creating E.164 domains.

## dns-enum-domain

dns-enum-domain - Configures ENUM domain in the DNS authoritative servers.

### Synopsis

```
dns-enum-domain <name> create
    [zone-template=<template-name>]
    [<nameserver>[person]]
dns-enum-domain <name> delete
dns-enum-domain <name> show
dns-enum-domain <name> get <attribute>
dns-enum-domain <name> set
    <attribute>=<value>
dns-enum-domain <name> unset <attribute>
dns-enum-domain list
dns-enum-domain listnames
dns-enum-domain listbrief

dns-enum-domain < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
dns-enum-domain < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
dns-enum-domain <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

This command is used to configure ENUM domain for the DNS servers.  
The DNS servers must be reloaded for changes to take effect.

### Examples

```
nrcmd> dns-enum-domain +123 create
nrcmd> dns-enum-domain +123 set serial=2
nrcmd> dns-enum-domain +123 show
```

### Status

## See Also

### Attributes

**description** [string](#)

Provides additional descriptive information.

**domain-prefix** [string](#) required,immutable

Specifies the E.164 number prefix for this domain.

**nameservers** [nlist](#)(obj(0))

Lists the nameservers for this domain.

**person** [dname](#)

Identifies the mailbox of the person responsible for this domain.

**serial** [rangeint](#)(1--1) default = 1

Sets the serial number for the domain. Note this value will only be applied to the SOA record if it is greater than the current serial number.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this domain.

**ttl** [dnsttl](#) default = 60m

Sets the time-to-live (ttl) value for the domain.

**view-id** [int](#) default = 0, immutable

Specifies the view identifier for this domain.

**zone-template** [nameref](#)(0)

Specifies the zone template to use when creating the associated zone.

## dns-enum-number

dns-enum-number - Configures ENUM number in the DNS authoritative servers.

### Synopsis

```
dns-enum-number <name> create
    <type> <subtype> <URI> [zone-template=
    <template-name>] [domain-prefix]
dns-enum-number <name> delete
dns-enum-number <name> show
dns-enum-number <name> get <attribute>
dns-enum-number <name> set
    <attribute>=<value>
dns-enum-number <name> unset <attribute>
dns-enum-number list
dns-enum-number listnames
dns-enum-number listbrief

dns-enum-number <name> addService <type>
    <subtype> <URI> [order [preference]]
dns-enum-number <name> removeService <type>
    <subtype> <URI>
dns-enum-number <name> listServices

dns-enum-number < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
dns-enum-number < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
dns-enum-number <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

This command is used to configure ENUM number for the DNS servers. The DNS servers must be reloaded for changes to take effect.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".

## Examples

```
nrcmd> dns-enum-number +123456 create tel tel tel:+19782523321
nrcmd> dns-enum-number +123456 show
```

## Status

## See Also

### Attributes

**description** [string](#)

Provides additional descriptive information.

**domain-prefix** [string](#) immutable

Specifies the E.164 number prefix for the parent domain.

**number** [string](#) required,immutable

Specifies the E.164 number for this entry.

**ported-nameserver** [dname](#)

Specifies the nameserver for a ported number that has been delegated.

**ported-number** [enumint](#)(no=0, ported-out=1, delegated-out=2, ported-in=3) default = no

Indicates whether this number has been ported to or from another carrier. Ported numbers can be:  
ported-out: Indicates the service for this number has been redirected to a new carrier.

delegated-out: Indicates the number has been delegated to a new carrier. The ported-nameserver attribute must also be set for this option.

ported-in: Indicates this number is in a range owned by another carrier. The domain-prefix should be set equal to the number for this case.

**services** [nlist](#)(obj(0)) required

Specifies the services for this number.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this domain.

**view-id** [int](#) default = 0, immutable

Specifies the view identifier for this domain.

**zone-template** [nameref](#)(0)

Specifies the zone template to use when creating the associated zone.

## dns-interface

dns-interface - Configures the DNS server's network interfaces

### Synopsis

```
dns-interface <name> create [<attribute>=<value>]
dns-interface <name> delete
dns-interface list
dns-interface listnames
dns-interface listbrief
dns-interface <name> show
dns-interface <name> set <attribute>=<value> [<attribute>=<value> ...]
dns-interface <name> get <attribute>
dns-interface <name> unset

dns-interface <name> enable <attribute>
dns-interface <name> disable <attribute>
```

## Description

The `dns-interface` command configures network interfaces for use by the Network Registrar DNS server. If there are no defined interfaces, the server discovers and uses all available interfaces on the system. When this list is present, the server uses only the available interfaces, if any, that match this list.

## Examples

## Status

## See Also

### Attributes

`address` [subnet](#)

Specifies the IP address and subnet mask of the DNS interface.

`ip6address` [prefix](#)

Specifies the IPv6 address and prefix length for one or more DNS interfaces.

`name` [string](#) required,unique

Specifies the user-assigned name of the DNS server interface.

## dns-update-map

`dns-update-map` - Configures a DNS update map of the DHCP and DNS server configurations needed to perform DNS updates

## Synopsis

```
dns-update-map <name> create <dhcp-servers> <dns-servers>
                                     <dns-config> [<attribute>=<value> ...]
dns-update-map <name> delete
dns-update-map list
dns-update-map listnames
dns-update-map listbrief
dns-update-map <name> show

dns-update-map <name> get <attribute>
dns-update-map <name> set <attribute>=<value> [<attribute>=<value> ...]
dns-update-map <name> unset <attribute>

dns-update-map <name> push [-report-only | -report]
```

## Description

The `dns-update-map` command lets you define and manage DNS update configuration maps. A DNS update map defines an update relationship between a DHCP policy and a list of DNS zones. The update map is designed to coordinate:

- DNS servers or Highly Available (HA) DNS server pairs
- DNS update ACLs or update policies
- DHCP servers or failover server pairs
- DHCP policy selection

An update map applies to all primary zones that the DNS server manages, and all scopes that the DHCP server manages.

The `push` command is only available when connected to a regional cluster.

## Examples

## Status

## See Also

[dhcp](#), [failover-pair](#), [dns](#), [ha-dns-pair](#), [dhcp-dns-update](#), [acl](#), [update-policy](#)

## Attributes

### dhcp-client-class [nameref](#)(0)

If `dhcp-policy-selector=use-client-class-policy`, this attribute specifies the named client-class embedded policy which should specify the `DnsUpdateConfig` referenced in this map. A new embedded policy will be created if one does not exist when the map is applied.

### dhcp-named-policy [nameref](#)(0)

If `dhcp-policy-selector=use-named-policy`, this attribute specifies the named policy which should specify the `DnsUpdateConfig` referenced in this map.

### dhcp-policy-selector [enumint](#)(`use-named-policy=1, use-client-class-embedded-policy=2, use-scope-embedded-policy=3`) default = `use-named-policy`

Indicates how to find the DHCP Policy to which to attach the `DnsUpdateConfig` referenced in the `dns-config` attribute. If a `named-policy` or `client-class-embedded-policy` is selected, the referenced policy and/or client-class must be preconfigured on the DHCP servers before the map is applied. If `scope-embedded-policy` is selected, it is applied to all scopes on the DHCP servers. A new embedded policy is created if one does not exist when the map is applied.

### dhcp-servers [oid](#) required

The DHCP server or DHCP failover pair associated with this configuration.

### dns-config [nameref](#)(0) required

The `DnsUpdateConfig` object associated with this configuration.

### dns-servers [oid](#) required

The DNS server or HA pair associated with this configuration.

### dns-update-acl [amelist](#)

The update-acl to apply to zones referenced by the `DnsUpdateConfig` in this map. If this attribute is set, then the value (if any) of the `dns-update-policy-list` attribute is ignored. If neither attribute is set, then a simple update-acl will be constructed enabling just the `dhcp-servers` to perform dns updates, using the IP address(es) of the single DHCP server or failover pair, and, if specified, the server-key from the `DnsUpdateConfig` referenced in the `dns-config` attribute.

### dns-update-policy-list [nlist](#)(obj(0))

The list of DNS update policies to apply to zones referenced by the `DnsUpdateConfig` in this map. If the attribute `dns-update-acl` is set, then this attribute is ignored. If neither attribute is set, then a simple update-acl will be constructed enabling just the `dhcp-servers` to perform dns updates, using the IP address(es) of the single DHCP server or failover pair, and, if specified, the server-key from the `DnsUpdateConfig` referenced in the `dns-config` attribute.

### name [string](#) required,unique

The name of this managed DNS update configuration.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## dns-view

`view` - Controls and configures DNS Views in the DNS Authoritative and Caching servers

NOTE: The `dns-view` command is a synonym for `view` command.

## Synopsis

```
view <name> create <attribute>=<value>
view <name> delete [-force]
view <name> show
view <name> get <attribute>
view <name> set <attribute>=<value>
view <name> unset <attribute>
view list
view listnames
view listbrief

view < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
view < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
view <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The view command is used to control and manage DNS Views for the DNS servers. The DNS servers must be reloaded for changes to take effect.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

```
nrcmd> view MyView create
nrcmd> view MyView set acl-match-clients=10.11.0.0/16
nrcmd> view MyView show
```

## Status

## See Also

### Attributes

**acl-match-clients** [amelist](#) default = any

Specifies the ACL that maps clients to this view based on the source address or TSIG key.

**description** [string](#)

An optional description for this DNS View configuration.

**name** [tag](#) required,unique

Specifies the name of the view.

**priority** [priority](#) default = 4294967295

Specifies the view priority relative to other views with common match clients. The lowest non-zero value has the highest priority. Views with common match clients should be assigned a unique non-zero value to ensure consistent behavior. A priority of 0 has the lowest possible priority, and should be reserved for the default view.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of the view.

**view-id** [int](#) immutable

Defines a unique integer identifier for the view that is assigned by the CCM server.

## dnssec

dnssec - Controls and configures DNSSEC processing in the Authoritative DNS server

## Synopsis

```
dnssec set <attribute>=<value> [<attribute>=<value>...]
dnssec unset <attribute>
dnssec get <attribute>
dnssec show

dnssec pull <cluster-name> [--report-only | -report]
dnssec push <cluster-list> [--report-only | -report]
```

## Description

The dnssec command lets you edit the Authoritative DNSSEC configuration. Use the zone command to enable DNSSEC for a primary zone on the authoritative DNS server.

The pull and push commands are only available when connected to a regional cluster. Push allows a list of clusters or "all".

## Examples

```
nrcmd> dnssec set zsk-algorithm=RSASHA1
```

## Status

## See Also

### Attributes

#### description [string](#)

A description of the authoritative DNSSEC configuration.

#### key-rollover [enumint](#)(Local=0, Regional=1) default = Local

Specifies whether the regional or local cluster should perform Zone Signing Key (ZSK) rollover. If using regional zone management, this setting should be set to regional in order to centrally manage key generation and rollover.

#### ksk-algorithm [enumint](#)(DSA=3, RSASHA1=5, RSASHA256=8, RSASHA512=10) default = RSASHA256

Specifies the cryptographic algorithm to be used for the Key Signing Key.

DSA	:	DSA/RSA-1,	value: 3,	range: 512-1024
RSASHA1	:	RSA/SHA-1,	value: 5,	range: 512-2048
RSASHA256	:	RSA/SHA-256,	value: 8,	range: 512-2048
RSASHA512	:	RSA/SHA-512,	value: 10,	range: 512-2048

#### ksk-bits [rangeint](#)(512-2048) default = 1024

Specifies the number of bits in the key and must be a multiple of 64. The value depends on the Key Signing Key algorithm (ksk-algorithm) chosen.

DSA	:	DSA/RSA-1,	value: 3,	range: 512-1024
RSASHA1	:	RSA/SHA-1,	value: 5,	range: 512-2048
RSASHA256	:	RSA/SHA-256,	value: 8,	range: 512-2048
RSASHA512	:	RSA/SHA-512,	value: 10,	range: 512-2048

#### ksk-rollover-interval [time](#) default = 4w2d

Specifies the time interval for the Key Signing Key (KSK) rollover process. It determines the lead time for the new key prior to the current key deactivation-date.

#### name [string](#) default = Default, unique,immutable

Specifies the name of authoritative DNSSEC configuration.

#### zsk-algorithm [enumint](#)(DSA=3, RSASHA1=5, RSASHA256=8, RSASHA512=10) default = RSASHA256

Specifies the cryptographic algorithm to be used for the Zone Signing Key.

DSA	:	DSA/RSA-1,	value: 3,	range: 512-1024
RSASHA1	:	RSA/SHA-1,	value: 5,	range: 512-2048

```
RSASHA256 : RSA/SHA-256, value: 8, range: 512-2048
RSASHA512 : RSA/SHA-512, value: 10, range: 512-2048
```

**zsk-bits** [rangeint](#)(512-2048) default = 1024

Specifies the number of bits in the key and must be a multiple of 64. The value depends on the Zone Signing Key algorithm (zsk-algorithm) chosen.

```
DSA : DSA/RSA-1, value: 3, range: 512-1024
RSASHA1 : RSA/SHA-1, value: 5, range: 512-2048
RSASHA256 : RSA/SHA-256, value: 8, range: 512-2048
RSASHA512 : RSA/SHA-512, value: 10, range: 512-2048
```

**zsk-lifetime** [time](#) default = 1y

Specifies the lifetime of a Zone Signing Key (ZSK). This defines the time interval where the key is used to sign zones. It is used to determine the deactivation-date when a ZSK key is created. The configured value MUST be greater than the zsk-rollover-interval. A value that is 10 times greater is recommended.

**zsk-rollover-interval** [time](#) default = 4w2d

Specifies the time interval for the Zone Signing Key (ZSK) rollover process. It determines the lead time for the new key prior to the current key deactivation-date. Configured interval should be more than maximum TTL of the zones plus the propagation delay, to avoid bogus zone information.

## dnssec-key

dnssec-key - Manage Authoritative DNSSEC Key objects

### Synopsis

```
dnssec-key <name> create [<attribute>=<value>...]
dnssec-key <name> delete [-force]
dnssec-key <name> show

dnssec-key <name> set <attribute>=<value>
                                     [<attribute>=<value>...]

dnssec-key <name> unset <attribute>
dnssec-key <name> get <attribute>

dnssec-key list
dnssec-key listnames
dnssec-key listbrief
dnssec-key getStatus

dnssec-key < <name> | all > pull < replace | exact > <cluster-name>
                                     [-report-only | -report]
dnssec-key < <name> | all > push < replace | exact > <cluster-list>
                                     [-report-only | -report]
dnssec-key <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The dnssec-key command creates and manages Authoritative Domain Name System Security Extensions (DNSSEC) keys for zone signing.

DNS Security Extensions (DNSSEC) provides protection against malicious or forged answers by adding digital signatures into DNS data, so each DNS response can be verified for integrity and authenticity. DNSSEC uses various algorithms to generate keys, which can be used do digitally sign the zones.

To configure DNSSEC protected zones, a user must first create a DNSSEC key, then sign the zone using the key.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".

### Examples

```
nrcmd> dnssec-key key-1 create algorithm=RSASHA1 bits=2048 type=ZSK
nrcmd> dnssec-key key-1 delete
```

### Status



## See Also

### Attributes

**activation-date** [date](#) immutable

Specifies the activation date and time for this key. Beginning at this date and time, the key will be used to sign RRSets.

**algorithm** [enumint](#)(DSA=3, RSASHA1=5, RSASHA256=8, RSASHA512=10) default = RSASHA256, immutable

Specifies the cryptographic algorithm to be used for the Key.  
DSA : DSA/RSA-1, value: 3, range: 512-1024  
RSASHA1 : RSA/SHA-1, value: 5, range: 512-2048  
RSASHA256 : RSA/SHA-256, value: 8, range: 512-2048  
RSASHA512 : RSA/SHA-512, value: 10, range: 512-2048

**bits** [rangeint](#)(512-2048) default = 1024, immutable

Specifies the number of bits in the key which must be a multiple of 64. The value depends on the Key algorithm (algorithm) chosen.  
DSA : DSA/RSA-1, value: 3, range: 512-1024  
RSASHA1 : RSA/SHA-1, value: 5, range: 512-2048  
RSASHA256 : RSA/SHA-256, value: 8, range: 512-2048  
RSASHA512 : RSA/SHA-512, value: 10, range: 512-2048

**deactivation-date** [date](#) default = none

Specifies the deactivation date and time for this key. Until this date and time, the key will be used to sign RRSets. This attribute must be 0 for Key Signing Keys. Key Signing Keys remain active until the key rollover process is initiated.

**description** [string](#)

An optional description of the authoritative DNSSEC key configuration.

**enable-signing** [bool](#) default = false

Indicates this key is enabled and will be used to sign the zones.

**expiration-date** [date](#) default = none

Specifies the date and time this Zone Signing Key is scheduled to be removed. If 0, automatic removal is disabled and the key must be deleted by user action. This attribute must be 0 for Key Signing Keys. Key Signing Keys remain active until the key rollover process is initiated. When the rollover process is complete, the key can be deleted by user action.

**key-group** [tag](#)

Specifies the name of the key group for this key. The allowed characters include a-z, A-Z, 0-9, ., -, \_, and +.

**name** [string](#) required,unique,immutable

Specifies the name of this key.

**rollover-due-date** [date](#) transient

Specifies the date and time, when this key should be (or was) rolled over. This transient attribute is used only for reporting.

**status** [string](#) transient

Specifies the current status of the key. This transient attribute is used only for reporting.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**type** [enumint](#)(ZSK=0, KSK=1) default = ZSK, immutable

Specifies whether the key is a Zone Signing Key (ZSK) or a Key Signing Key (KSK).

## exit

exit - Exits from the current configuration mode or nrcmd session.

## Synopsis

**exit**

## Description

When executed in global configuration mode, the **exit** command lets you exit the current nrcmd session. If you have unsaved changes, they will be flushed to the database before the session exits.

When executed in smart license configuration mode, the **exit** command brings you to global configuration mode.

## Examples

## Status

## See Also

[save](#)(nrcmd)

# export

**export** - exports configuration information to a file

## Synopsis

```
export leases [ -client | -server ]  
                [ -vpn <vpn-name> ]  
                [ -time-ascii | -time-numeric ] <filename>  
export zone <zone name> {-protected| -unprotected | -all} [<filename>]  
export zonenames {forward | reverse | both} [<filename>]  
export hostfile [<filename>]  
export keys <filename>  
export key <keyname> <filename>  
export option-set <option-set name> <filename>  
export dnssec-ds <zone name> <filename>  
export changeLog <filename> [<attribute>=<value> ...] [-all]
```

## Description

The export command lets you export data in one of several common formats.

```
export leases [ -client | -server ]  
                [ -vpn <vpn-name> ]  
                [ -time-ascii | -time-numeric ] <filename>  
Use the export leases command to export leases to a file.
```

If **-client** is specified (or **-server** is not specified), only leased leases are exported.

If **-server** is specified the export includes expired and leased leases.

The optional **time-ascii** and **time-numeric** keywords specify how to output date/time fields to the text file. The default is **time-ascii**.

If no **vpn** is specified, the current-vpn of the session is used.

When specifying a <vpn-name> to any export command which supports it, the name "global" (with or without the quotes) will specify the global (i.e., unnamed or default) vpn. The name "all" (also with or without the quotes) will specify that all vpn (including the global one) should be exported.

**export zone** <zone name> **{-protected | -unprotected | -all}** [<filename>]  
Use the export zone command to export the resource records in the specified zone. BIND can parse the format of the output.

**export zonenames** **{forward | reverse | both}** [<filename>]  
Use the export zonenames command to export the list of zones that match a given criteria.

**export hostfile** [<filename>]  
Use the export hostfile command to export the CCM DNS information in the cluster in a UNIX hostfile format.

**export keys** <filename>  
**export key** <keyname> <filename>  
You can use the export command to export the TSIG keys that are configured on a cluster with the export keys command. You can also specify to export a single key with the export key <keyname> command. These commands will generate key definitions in BIND syntax so they may be either imported into other clusters or BIND configurations.

**export option-set** <option-set name> <filename>  
The export option-set command writes out a text file that may be loaded into a running server with the option-set import command.

**export dnssec-ds** <zone name> <filename>  
Use the export dnssec-ds command to export the Delegation Signer (DS) record of a signed zone to the specified file. This command is applicable only for primary zones.

**export changeLog** <filename> [<attribute>=<value> ...] **[-all]**  
Exports the CCM change log to a CSV formatted file (specify a filename of - (hyphen) to output to the screen). If **-all** is not specified, only the most recent 1000 entries are reported. If **-all** is specified, all entries are reported. One or more of the following filter attributes can be specified to limit the reported data:

## Examples

## Status

## See Also

[session](#) current-vpn

## extension

extension - Integrates user-written DHCP extensions into the DHCP server

## Synopsis

```
extension list
extension listnames
extension listbrief
extension <name> create <lang> <file> <entry> [<attribute>=<value>...]
extension <name> delete
extension <name> get <attribute>
extension <name> set <attribute>=<value> [<attribute>=<value> ...]
extension <name> unset <attribute>
extension <name> show
```

## Description

The extension command lets you configure extension modules. To extend the DHCP server with an extension module, you must do the following:

1. Write the extension module in either Tcl, C or C++, and install it in the server scripts directory.
2. Configure the DHCP server to recognize this extension, using this command.
3. Attach the configured extension to one or more DHCP script points using the `dhcp attachExtension` command.

## Examples

Create a tcl script, `sample1`, that does something.  
`nrcmd> extension sample1 create tcl sample1.tcl sample_ext`

## Status

## See Also

[dhcp AttachExtension](#)

### Attributes

entry [string](#) required

Identifies the entry point for the module. This function is called from any extension point to which this module is bound.

file [string](#) required

Provides the filename relative to the directory extensions in the installation, or as an absolute pathname, but this cannot contain a sequence of two dots (`..`).

init-args [string](#)

Describes the arguments that should be passed to the init-entry point function. The arguments are passed in the environment dictionary using the key "Arguments".

init-entry [string](#)

Specifies the name of the init entry point. If set, the server calls this function when the server loads the module.

lang [enumstr](#)(Tcl=1, Dex=2)

Describes the language in which the script or module is implemented. 'Tcl' indicates that the module is a Tcl script (tcl8.4). 'Dex' indicates that the module is a shared object with C calling interfaces.

name [string](#) required,unique,immutable

Designates the script or module. The DHCP server uses this name to assign scripts or modules to script points.

## failover-pair

`failover-pair` - configures a DHCP failover relationship

## Synopsis

```
failover-pair <name> create <main-cluster/address>  
                                <backup-cluster/address>  
                                [<attribute>=<value> ...]
```

```
failover-pair <name> delete  
failover-pair list  
failover-pair listnames  
failover-pair listbrief  
failover-pair <name> show
```

```

failover-pair <name> get <attribute>
failover-pair <name> set <attribute>=<value> [<attribute>=<value> ...]
failover-pair <name> unset <attribute>

failover-pair <name> sync < update | complete | exact >
    [< main-to-backup | backup-to-main >]
    [-report-only | -report]

failover-pair <name> setPartnerDown [<date>]

failover-pair <name> getStatus

failover-pair <name> pollLeaseHistory
failover-pair <name> getLeaseHistoryState

failover-pair <name> rebalancePool

```

## Description

The failover-pair command lets you define and manage the failover relationship between a main and backup server.

Either the main and backup clusters or the main and backup server IP addresses can be specified with the create command. If the main-server and backup-server addresses are set, the cluster addresses will only be used for synchronization of the server configuration. The referenced clusters must be configured with appropriate connection credentials for the sync and pollLeaseHistory commands to be successful.

The pollLeaseHistory and getLeaseHistoryState commands are only available when connected to a regional cluster.

```
failover-pair <name> setPartnerDown [<date>]
```

The setPartnerDown command notifies the DHCP server that its failover partner server is down. The date specified represents a time equal to or later than the last known time the partner server could have been operational. If no date is specified, the last known time of contact with the partner server is used.

The partner server's available IP addresses will be available to the remaining server to lease once the MCLT has elapsed from the time that is used by the setPartnerDown command. These IP addresses will continue to be in state other-available, but the remaining server will be able to allocate them to clients once the MCLT has elapsed.

If the MCLT has not passed since the partner server went down, you can make that partner's leases available to the remaining server to lease to new clients immediately, though there is risk to this approach. To make these leases immediately available, you would enter a time on the setPartnerDown command which was more than the MCLT prior to the current time. This will allow the remaining server to immediately lease all available leases to DHCP clients. Should you do this, you will incur a risk that there will be some doubly allocated IP addresses, since some address allocations made by the partner server to DHCP clients may not have been communicated to the remaining server prior to it going down. These IP addresses may then be allocated to other DHCP clients by the remaining server. We do not recommend this approach -- but if you are out of addresses, it is certainly preferable to disabling failover altogether.

The time value should be entered using the local time of the nrcmd process. Formats for the date are:

```
-<num><value>
```

where <num> is a decimal number and <value> is one of 's', 'm', 'h', 'd', 'w', in which 's' is seconds, 'm' is minutes, 'h' is hours, 'd' is days and 'w' is weeks.

```
<month> <day> <hour>:<minute>[:<second>] <year>
```

where <month> is the name or first three letters of the name of the month, <hour> is the hour on a 24- hour clock, and <year> is the fully-specified year or a two-digit representation in which 98 = 1998, 99 = 1999 and all other two digit values XX = 20XX.

```
failover-pair <name> getStatus
```

This displays the failover-pair's related server object (same as dhcp getRelatedServers full, but without any other servers).

```
failover-pair <name> pollLeaseHistory
```

**failover-pair <name> getLeaseHistoryState**

These commands are only available when connected to a regional cluster and can be used to initiate a lease history poll operation or examine the lease history state.

**failover-pair <name> rebalancePool**

This command can be used to initiate a pool rebalance process.

## Examples

## Status

## See Also

[cluster](#)

## Attributes

**backup [oid](#)(0)**

Identifies the cluster that contains the backup server for a failover pair.

**backup-ip6address [ip6addr](#)**

Controls the IPv6 address used for the failover protocol on the backup server. If this value is unset, the address specified for the backup cluster is used. Cisco recommends setting this value only if the server is configured with different addresses for configuration management and clients requests. This value may be set to 0::0 to disable use of IPv6 for failover communication. If both IPv4 and IPv6 addresses are available, the servers will try both transports for the TCP connection and use whichever comes up first.

**backup-pct [percent](#) default = 50%**

Controls the percentage of available addresses that the main server sends to the backup server. Set this value on the main server. If it is set on a backup server, it is ignored (to enable copying of configurations). Unless you explicitly set this value on a scope and you disable load balancing, the value set here becomes the default value.

**backup-server [ipaddr](#)**

Controls the IPv4 address used for the failover protocol on the backup server. If this value is unset, the address specified for the backup cluster is used. Cisco recommends setting this value only if the server is configured with different interfaces for configuration management and clients requests. This value may be set to 0.0.0.0 to disable use of IPv4 for failover communication. If both IPv4 and IPv6 addresses are available, the servers will try both transports for the TCP connection and use whichever comes up first.

**dynamic-bootp-backup-pct [percent](#)**

Determines the percentage of available addresses that the main server sends to the backup server for scopes on which dynamic BOOTP is enabled. If defined, it must be defined on the main server. If it is defined in a backup server, it is ignored (to enable copying of configurations). If it is not defined at all or the value is 0, the "backup-pct" is used instead. This parameter is separate from "backup-pct" because if dynamic BOOTP is enabled on a scope, a server will never, even in PARTNER-DOWN state, grant leases on addresses that are available to the other server because they can never safely be assumed to be available again. The MCLT has no meaning for dynamic BOOTP leases.

**failover [bool](#) default = true**

Enables failover configuration. If you disable this attribute, you turn off failover on attached subnets without changing configuration fundamentals.

**load-balancing [bool](#) default = disabled**

Determines whether load balancing (RFC 3074) is enabled on a failover pair. The default is disabled. When enabled, the backup-pct is ignored and the main and backup server evenly split the client load and available leases for all scopes in the failover relationship (that is, as if backup-pct were configured at 50%).

**main [oid](#)(0)**

Identifies the cluster with the main server for a failover pair.

**main-ip6address** [ip6addr](#)

Controls the IPv6 address used for the failover protocol on the main server. If this value is unset, the address specified for the main cluster is used. Cisco recommends setting this value only if the server is configured with different addresses for configuration management and clients requests. This value may be set to 0::0 to disable use of IPv6 for failover communication. If both IPv4 and IPv6 addresses are available, the servers will try both transports for the TCP connection and use whichever comes up first.

**main-server** [ipaddr](#)

Controls the IPv4 address used for the failover protocol on the main server. If this value is unset, the address specified for the main cluster is used. Cisco recommends setting this value only if the server is configured with different interfaces for configuration management and clients requests. This value may be set to 0.0.0.0 to disable use of IPv4 for failover communication. If both IPv4 and IPv6 addresses are available, the servers will try both transports for the TCP connection and use whichever comes up first.

**mclt** [rangetime](#)(5m-1w) default = 60m

Sets the maximum client lead time in seconds. This attribute controls how far ahead of the backup server that you can make the client lease expiration. You must define this value on both the main and backup servers, and make sure the value is identical on both servers.

**name** [string](#) required,unique

Names a failover pair.

**rebind-limit** [time](#)

Sets the limit on the rebinding (T2) time beyond the renewal (T1) time that the failover partner that would not normally respond to a Request, Renew, or Rebind request uses when responding on behalf of the partner. When set (to a non-zero value), this expedites returning the client to being serviced by the failover partner that it should be serviced by. Note that for DHCPv6, this can also cause a server to drop a RENEW request it would have serviced based on the server-id option (if the failover partner is operational and should service the client).

**relay-health-check** [enumby](#) default = disabled

Sets whether health checking is enabled and for which protocols when failover communication is in the normal state. It can be set to disabled (default), v4-only, v6-only, or both.

**rhc-elapsed-time** [rangetime](#)(0-10m55s) default = 10s

Sets the minimum time a client must report in the DHCPv4 secs field or DHCPv6 elapsed time (8) option before the server will respond for its partner when the communication between the partner and relay is determined to be down. If set to 0 (not recommended), the server will respond to all requests.

**rhc-monitor-first-relay** [bool](#) default = false

Determines if DHCPv6 traffic should be monitored based on innermost (first) relay or outermost relay (from where DHCP Server receives the message). It is recommended to monitor traffic based on outermost relay, as it will reduce the number of relays to be monitored and hence reduce memory usage.

**rhc-request-count** [rangeint](#)(3-10000) default = 10

Sets the health check partner request count. This is the number of (DHCPv4) DHCPDISCOVER or (DHCPv6) Solicit requests the partner could have responded to before this server starts responding to client requests that it would normally not respond to.

**rhc-response-time** [rangetime](#)(0-60m) default = 0

Sets the health check response time (in seconds). This is the amount of time this server will respond to (DHCPv4) DHCPDISCOVER or (DHCPv6) Solicit requests on behalf of its partner as it assumes there are communication issues between the relay and that partner. This is to assure that this server does not respond forever if clients keep accepting offers from this server. If set to 0, this server will continue to respond to clients until it receives a (DHCPv4) DHCPREQUEST or (DHCPv6) Request packet addressed to another server.

**rhc-restart-time** [rangetime](#)(5s-60m) default = 30s

Sets the health check restart time (in seconds). When no (DHCPv4) DHCPDISCOVER or (DHCPv6) Solicit requests to which the partner would respond are received for this interval of time, the times and counts for the monitored relay will be restarted. This is to assure that the health check is based on very recent data.

#### rhc-server-preference [decimal-byte](#)

Sets the DHCPv6 server preference to use when responding to a Solicit on behalf of the failover partner. Only used if a server preference option has been configured and the rhc-server-preference value is lower.

#### rhc-unresponsive-time [rangetime](#)(5s-60m) default = 60s

Sets the health check unresponsive time (in seconds). This is the minimum amount of time over which this server must fail to receive a (DHCPv4) DHCPREQUEST or (DHCPv6) Request packet addressed to another server before it starts responding to requests on behalf of its partner as it assumes there are communication issues between the relay and that partner.

#### rhc-warning-interval [rangetime](#)(1s-60s) default = 30s

Sets the minimum interval between logging of warning messages when the server is responding for its partner because of likely communication issues between its partner and a relay agent.

#### safe-period [time](#) default = 4h

Controls the safe period, in seconds. It does not have to be the same on both main and backup servers. It only has meaning if use-safe-period is enabled.

#### scopetemplate [oid](#)

Associates a scope template with a specified failover pair.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### use-safe-period [bool](#) default = enabled

Controls whether a server can enter PARTNER-DOWN state without an operator command. If disabled, a server never enters PARTNER-DOWN without an operator command.

## group

group - Configures a named group of administrators

### Synopsis

```
group <name> create [<attribute>=<value>]
group <name> delete
group list
group listnames
group listbrief
group <name> show
group <name> set <attribute>=<value> [<attribute>=<value> ...]
group <name> get <attribute>
group <name> unset <attribute>

group <name> enable <attribute>
group <name> disable <attribute>

group < <name> | all > pull < ensure | replace > <cluster-name>
    [-report-only | -report]
group < <name> | all > push < ensure | replace | exact > <cluster-list>
    [-omitrelated] [-report-only | -report]
group <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The group command configures the specified group of administrators. Administrator groups are used to associate an admin to one or more roles that control access to operations and data.

The pull, push and reclaim commands are only available when connected to a regional cluster. The push and reclaim commands allow a list of clusters or "all".

The push operation will also push the related roles (using replace mode) and related owners and regions (using ensure mode) unless -omitrelated is specified to prevent this.

### Examples



## Status

## See Also

[admin](#), [role](#)

## Attributes

desc [string](#)

Describes this group.

name [string](#) required,unique

Names this group of administrator roles.

tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## gss-tsig

gss-tsig - configure a GSS-TSIG objects

## Synopsis

```
gss-tsig list
gss-tsig listnames
gss-tsig listbrief
gss-tsig <name> show
gss-tsig <name> create
    [<attribute>=<value>...]
gss-tsig <name> delete
gss-tsig <name> get <attribute>
gss-tsig <name> set <attribute>=<value>
    [<attribute>=<value> ...]
gss-tsig <name> unset <attribute>

gss-tsig < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
gss-tsig < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
gss-tsig <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The gss-tsig command creates and manages transaction signature (TSIG) for DNS updates and zone transfers in dynamically. GSS TSIG security, as defined in RFC 3645, enables both DNS and DHCP servers to authenticate DNS updates. GSS TSIG security uses the gss-tsig (RFC 2930, RFC 2743 and RFC 3645) algorithm to generate a signature that is used to authenticate the requests and responses. The GSS TSIG client and server use the TKEY and TSIG resource records to provide security services.

GSS-TSIG object needs to be created and associated in the DHCP/DNS server object to change the default server behaviour in managing GSS keys.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".

## Examples

## Status

## See Also

### Attributes

**description** [string](#)

An optional description for this gss-tsig configuration object.

**name** [string](#) required,unique

Identifies the name of the gss-tsig configuration object.

**tkey-max-exchanges** [rangeint](#)(2-10) default = 5

The maximum number of times that a TKEY RRs will be exchanged between a client and the server during a particular key negotiation to prevent endless looping as per RFC 2930. Default value is 5.

**tkey-session-time** [rangetime](#)(0-1y) default = 5m

Specifies the user configurable maximum lifetime of a negotiated TKEY. Lifetime of a negotiated TKEY can be controlled by two factors. The first being by the TKEY expiry time given by the Kerberos server which is obtained during the initial TKEY negotiation. The second being through this attribute. If set to 0, this attribute is disabled and the TKEY lifetime is controlled only by the Kerberos TKEY expiry time obtained during the TKEY negotiation. When this attribute is configured with a value > 0, the minimum of Kerberos TKEY expiry time and this value is taken as the maximum lifetime of the TKEY. Default value is 300 seconds.

**tkey-table-max-size** [rangeint](#)(10-65535) default = 32767

The server and client will maintain some required data in TKEY table when performing TKEY negotiation. This attribute bounds the TKEY table by defining the maximum number of key records. The new TKEY query negotiation will fail when TKEY table hit this maximum size. Default size is 32767.

**tkey-table-purge-interval** [rangetime](#)(10s-10m) default = 60s

This attribute will define the interval to purge expired key records in TKEY table. Default value is 60 sec.

## ha-dns-pair

ha-dns-pair - configure a High Availability DNS relationship

### Synopsis

```
ha-dns-pair <name> create <main-cluster/address>
                    <backup-cluster/address>
                    [<attribute>=<value> ...]

ha-dns-pair <name> delete
ha-dns-pair list
ha-dns-pair listnames
ha-dns-pair listbrief
ha-dns-pair <name> show

ha-dns-pair <name> get <attribute>
ha-dns-pair <name> set <attribute>=<value> [<attribute>=<value> ...]
ha-dns-pair <name> unset <attribute>

ha-dns-pair <name> sync < update | complete | exact >
                    < main-to-backup | backup-to-main > [from-regional]
                    [-report-only | -report]

ha-dns-pair <name> getstatus [full]
```

### Description

The ha-dns-pair command lets you define and manage the High Availability relationship between a main and backup DNS server.

Either the main and backup clusters or the main and backup server IP addresses can be specified with the create command. If the ha-dns-main-server and ha-dns-backup-server addresses are set, the cluster addresses will only be used for synchronization of the server configuration. The referenced clusters must be configured

with appropriate connection credentials for the sync command to be successful.

Note: When running in local mode, the 'from-regional' sync option does not apply. Regardless of the synchronization option (from-regional, main-to-backup, backup-to-main), properties set on the ha-dns-pair will always replace values present on the DNS server object.

## Examples

## Status

## See Also

[cluster](#)

## Attributes

### backup [oid](#)(0)

The cluster reference for the backup server in this DNS HA pair relationship.

### description [string](#)

An optional description for this HA-DNS configuration.

### ha-dns [bool](#) default = enabled

This attribute enables/disables HA on the DNS server.

### ha-dns-backup-address [ipaddr](#)

The IP address to use for the HA DNS protocol on the backup server. If this value is unset, the address specified for the backup cluster will be used. In general, it should only be set if the server is configured with different interfaces for configuration management and update requests. The HA DNS protocol should always be configured with the interface used to service updates.

### ha-dns-backup-ip6address [ip6addr](#)

Specifies the IPv6 address to use for the HA DNS protocol on the backup server. If this value is unset, the address specified for the backup cluster will be used. In general, it should only be set if the server is configured with different interfaces for configuration management and update requests. The HA DNS protocol should always be configured with the interface used to service updates. If both IPv4 and IPv6 addresses are configured, the IPv6 address will be used.

### ha-dns-backup-mname [dname](#)

Specifies the DNS name of the HA backup server. When set and ha-dns-soa-mname-update is enabled, the SOA MNAME field for all HA primary zones is set to this FQDN when the HA backup server is accepting updates. DNS clients can perform SOA queries to determine which server should handle update requests. The SOA MNAME change is also propagated to secondary servers via zone transfers and notifies.

### ha-dns-main-address [ipaddr](#)

The IP address to use for the HA DNS protocol on the main server. If this value is unset, the address specified for the main cluster will be used. In general, it should only be set if the server is configured with different interfaces for configuration management and update requests. The HA DNS protocol should always be configured with the interface used to service updates.

### ha-dns-main-ip6address [ip6addr](#)

Specifies the IPv6 address to use for the HA DNS protocol on the main server. If this value is unset, the address specified for the main cluster will be used. In general, it should only be set if the server is configured with different interfaces for configuration management and update requests. The HA DNS protocol should always be configured with the interface used to service updates. If both IPv4 and IPv6 addresses are configured, the IPv6 addresses will be used.

### ha-dns-main-mname [dname](#)

Specifies the DNS name of the HA main server. When set and ha-dns-soa-mname-update is enabled, the SOA MNAME field for all HA primary zones is set to this FQDN when the HA main

server is accepting updates.  
DNS clients can perform SOA queries to determine which server should handle update requests. The SOA MNAME change is also propagated to secondary servers via zone transfers and notifies.

**ha-dns-port** [int](#) default = 653

This attribute specifies the TCP port number used for HA servers communication. The value of this attribute must be consistent for both HA servers.

**ha-dns-soa-mname-update** [bool](#) default = disabled

Controls whether or not the HA DNS servers will update the MNAME field of the SOA records on their primary HA zones when failing over. The `ha-dns-main-mname` and `ha-dns-backup-mname` attributes must also be set for the change to occur.

**ha-dns-sync-quiet-period** [time](#) default = 5m

Specifies the maximum amount of time a DNS HA server will wait before allowing non-HA traffic during synchronization. During the quiet period, all non-HA traffic is dropped to allow the zones to synchronize as quickly as possible. Once the synchronization is complete or the server exceeds the quiet period, non-HA traffic can resume. A value of 0 disables this setting.

**main** [oid](#)(0)

The cluster reference for the main server in this DNS HA pair relationship.

**name** [string](#) required,unique

The name of the DNS HA pair relationship.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## help

help - provides online help

### Synopsis

```
help
help <cmd> [<section> ...]
```

### Description

The help command provides online help.

If you type help without arguments, it displays a list of commands. If you type help with an argument, it displays the man page information for the command with that name.

You can select the sections of the man page output by specifying the section names after the 'help <cmd>' command. The section names are:

```
SYNOPSIS      the valid syntax for the command
DESCRIPTION  a textual description of the command behavior
EXAMPLES     examples of using the command
PROPERTIES   description of the attributes associated with the command
STATUS       description of the status codes returned by this command
```

For an introduction to nrcmd commands, use "help intro".

### Examples

```
nrcmd> help
100 Ok
... displays the list of commands
```

```
nrcmd> help dns
100 Ok
... displays the contents of the dns page
```

```
nrcmd> help dns synopsis
100 Ok
SYNOPSIS
dns
```

help <cmd> [<section> ...]

## Status

## See Also

[intro](#)

# import

import - loads server configuration information from a file

## Synopsis

```
import keys <file>
import leases <file>
import named.boot <file>
import named.conf <file>
import option-set <file>
```

## Description

The import command lets you import lease information into the DHCP server configuration or BIND configuration information into the DNS server configuration.

### import leases <file>

Before you can import leases, you need to perform several configuration steps:

1. Configure scopes in the DHCP server for the leases that are going to be imported. (see the scope command.)
2. If the host names for the leases are going to be dynamically entered into DNS as part of the import, configure zones in the DNS server to allow dynamic updates. (see the zone command)
3. Set the DHCP server to import mode so that it will not respond to other lease requests during the lease importing. (see the dhcp command)

After the leases have been imported, take the DHCP server out of import mode so that it will respond to other lease requests.

### import named.boot <file>

Imports an existing BIND 4.x.x configuration into DNS by parsing the BIND named.boot file and reading the zone data from the associated BIND zone files.

### import named.conf <file> <protected | unprotected>

Imports an existing BIND 8 or BIND 9 configuration into DNS by parsing the BIND named.conf file and reading the zone data from the associated BIND zone files. If no name-protection option is specified, the default is protected.

### import keys

Imports TSIG keys into the Cluster configuration by reading in key data from a file. This file can be generated by running the key generator utility (cnr keygen). The keys are written in BIND syntax and therefore can also be copied from a valid BIND configuration.

### import option-set

Imports an option-set from specified file. Compatible files can be generated using the 'export option-set' command.

## Examples

## Status

## See Also

# key

key - Manage TSIG key objects

## Synopsis

```
key list
key listnames
key listbrief
key <name> show
key <name> create <secret> [<attribute>=<value>...]
key <name> delete
key <name> get <attribute>
key <name> set <attribute>=<value> [<attribute>=<value> ...]
key <name> unset <attribute>

key < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
key < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
key <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The key command creates and manages transaction signature (TSIG) keys for DNS updates, zone transfers, queries, and recursions. TSIG security, as defined in RFC 2845, enables both DNS and DHCP servers to authenticate DNS updates. TSIG security uses the HMAC-MD5 (or keyed MD5) algorithm to generate a signature that is used to authenticate the requests and responses. The DHCP server uses TSIG keys to create TSIG resource records while processing DNS updates.

To configure TSIG security on a DHCP server, you must first create a shared key, then enable DNS update for your scopes by setting the dynamic-dns attribute to update-all). Also, enable the dynamic-dns-tsig attribute for forward or reverse zones for the scope or on the server level.

The pull, push and reclaim commands are only available when connected to a regional cluster. Push and reclaim allow a list of clusters or "all".

## Examples

## Status

## See Also

[export key](#), [export keys](#)

## Attributes

**algorithm** [enumstr](#)(hmac-md5=1) default = hmac-md5

The algorithm that this key is used with. Currently we only support hmac-md5.

**description** [string](#)

An optional description for this Key configuration.

**id** [int](#)

Displays an integer id for the key.

**secret** [key](#) required

A base64 encoded string used for transaction authentication.

**security-type** [enumstr](#)(TSIG=1) default = TSIG

The type of security that this key is going to be used for. Currently we only support TSIG keys.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**time-skew** [rangetime](#)(1s-60m) default = 5m

The time stamp fudge factor (amount that the time values can differ).

## ldap

ldap - Specifies the LDAP remote server's properties

### Synopsis

```
ldap list
ldap listnames
ldap listbrief
ldap <name> create <hostname> [<attribute>=<value>...]
ldap <name> delete
ldap <name> get <attribute>
ldap <name> set <attribute>=<value> [<attribute>=<value> ...]
ldap <name> unset <attribute>

ldap <name> disable <attribute>
ldap <name> enable <attribute>
ldap <name> show

ldap <name> setEntry <dictionary> <key>=<value>
ldap <name> getEntry <dictionary> <key>
ldap <name> unsetEntry <dictionary> <key>
```

### Description

The ldap command configures the LDAP servers that the DHCP server should communicate with. The DHCP server can read client configuration information from or write lease information to an LDAP enabled directory.

Use the setEntry, getEntry, and unsetEntry commands to set, query, and clear elements of the various dictionary properties in the LDAP server configuration. These dictionary properties provide a convenient mapping from strings keys to string values.

The dictionary values for the setEntry command are:

```
create-dictionary
create-string-dictionary
env-dictionary
query-dictionary
update-dictionary
v6-create-dictionary
v6-create-string-dictionary
v6-update-dictionary
```

### Examples

```
nrcmd> ldap ldap-1 create ldap.example.com port=389
nrcmd> ldap ldap-1 set username="cn=joe,o=Example Corp,c=US" password=access
nrcmd> ldap ldap-1 set search-path="o=Example Corp,c=US" search-scope=SUBTREE
nrcmd> ldap ldap-1 set search-filter=(cn=%s)
nrcmd> ldap ldap-1 setEntry query-dictionary cn=host-name
nrcmd> ldap ldap-1 setEntry query-dictionary sn=reserved-addresses
nrcmd> dhcp enable client-class
nrcmd> dhcp enable use-ldap-client-data
nrcmd> dhcp reload
```

## Status

## See Also

### Attributes

**can-create** [bool](#) default = disabled

Controls whether a particular LDAP server can create new entries to use to store lease state updates by DHCP server. See the create properties: `create-object-classes`, `v6-create-dictionary`, `v6-create-string-dictionary`. For creating IPv4 leases, `create-dictionary` is used whereas for creating IPv6 leases, `v6-create-dictionary` is used. This attribute also controls whether DHCP server should delete IPv6 leases on LDAP server when the lease gets deleted on DHCP server. DHCP server deletes lease on LDAP server only if it is configured to create one.

**can-query** [bool](#) default = disabled

Controls whether a particular LDAP server can be used for client queries. See the query properties: `env-dictionary`, `query-dictionary`, `search-attribute`, `search-filter`, `search-path`, and `search-scope`.

**can-update** [bool](#) default = disabled

Controls whether a particular LDAP server can be used to store lease state updates by DHCP server. See the update properties: `update-dictionary`, `update-search-attribute`, `update-search-filter`, `update-search-path`, `update-search-scope`, `v6-update-dictionary`, `v6-update-search-attribute`. For updating IPv4 leases, `update-dictionary`, `update-search-attribute` and `dn-attribute` are used whereas for updating IPv6 leases, `v6-update-dictionary`, `v6-update-search-attribute` and `v6-dn-attribute` are used.

**connections** [int](#) default = 1

Determines the number of connections that the server can make to an LDAP object. Network Registrar creates one thread for each connection configured in an LDAP object, and each thread can have a maximum of LDAP requests associated with its request queue. This is primarily a performance-tuning attribute. In some cases, having more than one connection can improve overall throughput.

**create-dictionary** [dict](#)

Maps LDAP attributes to DHCP lease or client attributes. If an entry does not exist and needs to be created, entries in this dictionary are set to the value of its corresponding DHCP lease or client attribute.

**create-object-classes** [string](#)

If `can-create` is enabled, specifies the object classes from which a new entry inherits.

**create-string-dictionary** [dict](#)

Maps LDAP attributes to user specified strings. If an entry does not exist and needs to be created, entries in this dictionary are set to the matching string.

**default-attribute-value** [string](#) default =

Provides a default attribute value (a string) to insert in any LDAP attribute whose associated lease attribute values are not present in the lease.

**dn-attribute** [string](#)

Determines how the server constructs the distinguished name (DN) of the LDAP entry to update or create. If the server can use one of the lease or client attributes, it formats the specified `dn-attribute` using the `dn-format` string to construct the object filter that specifies the LDAP server to modify.

**dn-create-format** [string](#)

Provides the distinguished name (DN) for entry creation. A `%` is required at the entry level and is replaced by the value of the `dn-attribute`. If you can construct the DN of the LDAP object created from one of the lease's or client's attributes, the server formats the specified `dn-attribute` using the `dn-format` string.

**dn-format** [string](#)

If the DN of the ldap object that is to be updated can be constructed from one of the lease's or client's attributes, the specified `dn-attribute` will be formatted using the



dn-format string to construct the query filter.

**enabled** [bool](#) default = true

Enables or disables this LDAP remote server.  
Prevents DHCP from attempting to use an LDAP server that is known to be unavailable.

**env-dictionary** [dict](#)

Specifies the environment dictionary that allows the server to retrieve additional LDAP attributes along with client-entry attributes. If any of these are present in a query's results, their values are made available to scripts through the request's environment dictionary. The LDA{ value is keyed by the value in the LDAP query env-dictionary.

**hostname** [string](#) required

Sets the hostname, IPv4 or IPv6 address of the server to connect to.

**limit-requests** [bool](#) default = enabled

Controls whether there should be a limit on the number of outstanding queries on each LDAP client connection. See the limit property: max-requests.

**max-referrals** [int](#) default = 0

Limits the number of LDAP referrals the server follows when querying. A value of zero prohibits following referrals.

**max-requests** [int](#) default = 20

Controls the number of outstanding queries.  
If you have set the 'limit-requests' feature to TRUE, any single LDAP connection will limit the number of outstanding queries to 'max-requests.' You can improve performance by limiting the number of outstanding queries.

**name** [string](#) required,unique

An arbitrary name used to refer to an individual server.

**password** [clrtxt](#)

Sets the password of a user with access to the parts of the directory that DHCP uses. Because you can configure LDAP servers to allow anonymous access, this is optional.

**password-encrypt** [encdata](#)

Specifies the encrypted password of the user.

**port** [int](#)

Specifies the port on the remote server to connect to.

**preference** [int](#) default = 1

Specifies the preference order in which LDAP servers are used. A positive integer greater than or equal to one. One (1) is the highest preference value.

**query-dictionary** [dict](#)

Maps LDAP attributes and DHCP attribute names. The server attempts to retrieve all LDAP attributes specified in the dictionary. When a query succeeds, the values for any ldap attributes that it returns are set in the corresponding client-entry attribute.

**query-timeout** [time](#) default = 3s

Specifies the number of seconds the DHCP server waits for a response to individual LDAP Query requests. After a query request times out, the DHCP server will drop the request and not process it again on another LDAP connection or LDAP Server. A query-timeout value of 3 seconds is a good value.  
Note: The timeout attribute configures the timeout for LDAP Update and Create requests.

**referral-attr** [string](#)

Indicates whether an LDAP response is a referral. The referral may or may not contain the DN for this query. If the DN is present (the default), the server uses it as the search path, along with a wildcard search-scope in the query that follows the referral. If not, the server builds the search path by formatting the data in the referral attribute with the referral-filter, using the existing search scope.

**referral-filter** [string](#)

In the absence of a distinguished name (DN), controls how a server formats referral-attr data. In such cases, the server formats the referral attribute's data with this filter expression to build a search path that uses the existing search-scope for the LDAP server.

**search-filter** [string](#)

Specifies the filter to apply in the client-entry query. The server formats the client's MAC address using the filter to specify the object that contains the client-entry data.

**search-path** [string](#)

Designates an object in the directory to use as a query starting-point. Together, the path and the search-scope control the portion of the directory that the server will search.

**search-scope** [enumint](#)(BASE=0, ONELEVEL=1, SUBTREE=2) default = SUBTREE

Controls the comprehensiveness of a search:  
If you specify the scope to be SUBTREE, the server searches all the children of the searchpath.  
If you specify the scope to be ONELEVEL, the server searches only the immediate children of the base object.  
If you specify the scope to be BASE, the server searches only the base object itself.

**threadwaittime** [mslime](#) default = 100

Sets the number of milliseconds that an LDAP client connection polls for the results of outstanding queries or updates.

**timeout** [time](#) default = 10s

Controls the number of seconds the DHCP server waits for a response to an individual LDAP update or create request. If an LDAP request times out, the DHCP server resubmits it to other LDAP connections. Further, if the DHCP server receives no response (that is, a result for an LDAP update or create) from an LDAP connection for the timeout seconds, DHCP marks this LDAP connection as 'Inactive' and tears down the connection, then reconnects. A timeout value of 10 seconds is a good value for LDAP create and update operations.  
Note: You can configure a separate timeout for LDAP query operations using the query-timeout attribute.

**update-dictionary** [dict](#)

Maps LDAP attributes to DHCP lease or client attributes. When an LDAP object is modified, each LDAP attribute that is present in this dictionary is set to the value of its corresponding DHCP lease or client attribute.

**update-search-attribute** [string](#)

If the DN of the object to be updated cannot be determined directly, the server must issue a query to retrieve the DN. In that case, the server uses data in the lease's or client's 'update-search-attribute,' and formats it using the 'update-search-filter' expression.

**update-search-filter** [string](#)

If the DN of the object to be updated cannot be determined directly, the server must issue a query to retrieve the DN. In that case, the server uses data in the lease's or client's 'update-search-attribute,' and formats it using the 'update-search-filter' expression.

**update-search-path** [string](#)

Determines the starting point for the portion of the directory containing LDAP objects for the server to update.

**update-search-scope** [enumint](#)(BASE=0, ONELEVEL=1, SUBTREE=2)

With update-search-path, controls the portion of the directory that contains the objects to be updated.  
The scope can be SUBTREE (includes all children of the searchpath), ONELEVEL (includes only the immediate children of the base object), or BASE (includes only the base object itself).

**username** [string](#)

Designates a user with access to the parts of the directory that DHCP uses. Because you can configure LDAP servers to allow anonymous access, this is optional.

**v6-create-dictionary** [dict](#)

Maps LDAP attributes to DHCP lease6 attributes. If an entry does not exist and needs to be created, entries in this dictionary are set to the value of its corresponding DHCP lease6 attribute.

**v6-create-string-dictionary** [dict](#)

Maps LDAP attributes to user specified strings for ipv6 lease. If an entry does not exist and needs to be created, entries in this dictionary are set to the matching string.  
If 'v6-create-string-dictionary' is not configured, 'create-string-dictionary' is used instead for ipv6 lease as well. Hence if string attributes are same for both ipv4 and ipv6, no need to configure both 'v6-create-string-dictionary' and 'create-string-dictionary'. Just configuring 'create-string-dictionary' is enough.

## v6-dn-attribute [string](#)

Determines how the server constructs the distinguished name (DN) of the LDAP entry to update or create. If the server can use one of the lease6 attributes, it formats the specified dn-attribute using the dn-format string to construct the object filter that specifies the LDAP server to modify.

## v6-update-dictionary [dict](#)

Maps LDAP attributes to DHCP lease6 attributes. When an LDAP object is modified, each LDAP attribute that is present in this dictionary is set to the value of its corresponding DHCP lease6 attribute.

## v6-update-search-attribute [string](#)

If the DN of the lease6 object to be updated cannot be determined directly, the DHCP server must issue a query to retrieve the DN. In that case, the DHCP server uses data in the lease's 'search-attribute,' and formats it using the 'update-search-filter' expression.

# lease

lease - Manage DHCP lease objects

## Synopsis

```
lease list [-macaddr=<mac-addr>] [-cm-macaddr=<mac-addr>]
           [-reservation-lookup-key=<lookup-key> [-mac|-blob|-string]]
           [-vpn=<vpn-name>] [-count-only]
lease listbrief [-macaddr=<mac-addr>] [-cm-macaddr=<mac-addr>]
               [-reservation-lookup-key=<lookup-key> [-mac|-blob|-string]]
               [-vpn=<vpn-name>] [-count-only]
lease listbrief [-vpn=<vpn-name>] [-count-only]
lease list -macaddr=<mac-address> [-vpn=<vpn-name>]
lease list -subnet <ip-address> <mask>
lease list -lansegment <ip-address> <mask>
lease [<vpn-name>/]<ip-address> [show]

lease [<vpn-name>/]<ip-address> get <attribute>

lease [<vpn-name>/]<ip-address> activate
lease [<vpn-name>/]<ip-address> deactivate
lease [<vpn-name>/]<ip-address> force-available

lease [<vpn-name>/]<ip-address> macaddr
lease [<vpn-name>/]<ip-address> get-scope-name
```

## Description

The lease command lets you view and manipulate current DHCP leases in the cluster.

When you specify the lease on which one of these commands is to operate, you may optionally specify a <vpn-name> in which the <ip-address> is to be found. You may specify the name of a currently defined vpn as the <vpn-name>, or use the reserved vpn name "global" (without the quotes) to specify the operation on leases which are not in any explicitly defined vpn. If you do not specify a <vpn-name>, then the session's current-vpn is used as a default.

```
lease list [-vpn=<vpn-name>] [-count-only]
lease list -macaddr <mac-address> [-vpn=<vpn-name>]
lease list -subnet <ip-address> <mask>
lease list -lansegment <ip-address> <mask>
lease [<vpn-name>/]<ip-address> [show]
```

The list command lists leases in the DHCP server. Only the leases in the current-vpn or specified vpn-name are listed. <vpn-name> may be "all" (without the quotes) to request leases in all vpns. If -count-only is specified, only the count of the number of leases is returned (no leases are displayed).

The list -subnet command lists all leases in a subnet (scopes whose address and mask match the query).

The list -lansegment command lists all leases in a LAN segment, meaning all leases in scopes whose address and mask match the query, as well as leases in secondary scopes whose primary scope's address and mask match the query.

The list `-macaddr` command lists all leases that are associated with the specified MAC address.

**lease** [<vpn-name>/]<ip-address> **activate**

**lease** [<vpn-name>/]<ip-address> **deactivate**

The activate and deactivate commands tell the DHCP server to make the specified lease active or inactive. An inactive lease is not given out, even if it is in the available state. Making a currently leased lease inactive does not affect its behavior until it has expired and become available again.

**lease** [<vpn-name>/]<ip-address> **force-available**

The force-available command forces the specified lease into the available state.

**lease** [<vpn-name>/]<ip-address> **macaddr**

The macaddr command provides the MAC address associated with the specified lease.

**lease** [<vpn-name>/]<ip-address> **get-scope-name**

The get-scope-name command provides the scope to which the lease belongs.

## Examples

## Status

## See Also

[export](#) leases, [session](#) current-vpn

## Attributes

**address** [ipaddr](#)

Specifies the IP address of the lease. The address is added at creation.

**binding-end-time** [date](#)

Within the DHCPlease database, holds the time at which a lease binding ended.

**binding-start-time** [date](#)

Within the DHCP lease database, shows the time at which a lease binding began.

**client-binary-client-id** [blob](#)

Displays the binary form of the client's client-identifier, if any.

**client-dns-name** [string](#)

Displays the client DNS name, which the DHCP server attempted (possibly successfully) to enter into the DNS server for a specified client. This attribute is related to the client-host-name, but may not be identical due to name collisions in the DNS server database.

**client-domain-name** [string](#)

Displays the domain (if any) to which the client DNS name belongs.

**client-duid** [blob](#)

Identifies the DUID of the RFC 4361 client identifier.

**client-flags** [flags](#)(client-valid=1, client-id-created-from-mac-address=2, client-dns-name-up-to-date=3, client-up-to-date-in-mcd=4, reverse-dns-up-to-date=5, host-name-has-changed=6, clear-host-name=7, domain-name-has-changed=8, dns-update-pending=9, client-fqdn-present=10, client-updates-name=11, use-test-before-update=12, avoid-dns-retry=13, dual-zone-dns-update=14, client-invalid-due-to-macaddress=15, in-limitation-list=16, used-over-limit-client-class=17, synthesized-dns-name=18, reservation-uses-client-id=19, client-id-from-override-id=20, client-id-from-string=21, override-client-id-from-string=22, limit-retention=23, deallocate-pri=24, renewal-tracked=25)

Displays any of the following values associated with the client lease state:

- 2 client-id-created-from-mac-address  
Indicates that the client-id was created for internal use from the client supplied MAC address. It is never reported externally if this is true.
- 3 client-dns-name-up-to-date

Indicates that the client-dns-name (A) is actually current in the DNS server database.

5 reverse-dns-up-to-date  
Indicates that the reverse (PTR) DNS entry is current in the DNS database.

9 dns-update-pending  
Indicates that a DNS operation is pending for this client.

16 in-limitation-list  
Indicates that this lease is presently in a limitation list using the limitation-id shown.

23 limit-retention  
Indicates if this lease is subject to lease time retention restrictions.

25 renewal-tracked  
Indicates if the client's renewal is being tracked (only if DHCP server's distribute-renewal feature is enabled).

These are for internal use only:

client-valid  
client-fqdn-present  
client-updates-name  
clear-host-name  
host-name-has-changed  
domain-name-has-changed  
use-test-before-update  
avoid-dns-retry (no longer used)  
dual-zone-dns-update  
client-invalid-due-to-macaddress  
used-over-limit-client-class  
synthesize-dns-name  
reservation-uses-client-id  
client-id-from-override-id  
client-id-from-string  
override-client-id-from-string  
deallocate-prl

#### client-host-name [string](#)

Displays the DNS name that the client requested the DHCP server to place in the DNS server.

#### client-iaid [int](#) default = 0

Identifies the IAID of the RFC 4361 client identifier.

#### client-last-transaction-time [date](#)

Displays the time when the client most recently contacted the DHCP server.

#### client-mac-addr [macaddr](#)

Displays the MAC address which the client presented to the DHCP server.

#### client-os-type [string](#)

Indicates the operating system of the client. This attribute is used only by the updateSms keyword and has no other purpose. If you enable failover, the main server transmits this value to the backup server. The syntax of this attribute's value is OS-name major.minor. Operating system values are as follows:

Microsoft Windows NT Server  
Microsoft Windows NT Advanced Server  
Microsoft Windows NT Workstation 4.0  
Microsoft Windows NT Workstation 3.51  
Microsoft Windows 2000 Professional  
Microsoft Windows 95  
Microsoft Windows 9x  
Microsoft Windows for Workgroups  
Microsoft Windows  
Dos  
Netware  
LANMAN Workstation  
LANMAN Server  
OS/2  
MAC OS

#### client-override-client-id [blob](#)

The value of the override-client-id expression for this client. If it appears, it is used as the client-id for this client.

#### client-vendor-class [option](#)

The most recently received client vendor class option.

#### client-vendor-info [option](#)

The most recently received vendor-specific information option.

#### data-source [enumint](#)(unknown=0, main-main=4, backup-main=5, main-backup=6, backup-backup=7, main-main-active=20, backup-main-active=21, main-backup-active=22, backup-backup-active=23, main-main-history=28, backup-main-history=29, main-backup-history=30, backup-backup-history=31)

Records the original source of the lease data and the machine from which the data was retrieved.

0 unknown  
4 main-main  
20 main-main-active  
28 main-main-history  
Indicates the data originated on the main server and was retrieved from the main server.  
5 backup-main  
21 backup-main-active  
29 backup-main-history

```
        Indicates the data originated on the backup server and
        was retrieved from the main server.
6      main-backup
22     main-backup-active
30     main-backup-history
        Indicates the data originated on the main server and
        was retrieved from the backup server.
7      backup-backup
23     backup-backup-active
31     backup-backup-history
        Indicates the data originated on the backup server and
        was retrieved from the backup.
        The suffix -active denotes the data was returned from
        the active portion of the lease-state database while
        -history indicates that the data was from the history
        portion of the lease-state database.
When viewing leases with the UI's, you will see all four
values routinely, especially if load-balancing is enabled.
When looking at lease history records, main-main and
backup-backup are the usual values, but in cases where the
lease history poller has determined that some data may be
missing, then main-backup and backup-main can appear as
well.
```

#### expiration [date](#)

Displays the date and time the lease will expire.

#### failover-expiration-time [date](#)

The latest time that this server has ever acked to the failover partner. In the absence of information from the failover partner, this server must wait for this time to expire before it can act on the state-expiration-time.

#### flags [flags](#)(reserved=1, deactivated=3, failover-updated=5, dynamic=7, client-reserved=13, use-bndack-time=15)

Displays flags that describe this lease:

```
1      (lease) reserved
        The lease is reserved because of a lease reservation object.
        The reservation-lookup-key specifies the client for which this
        lease is reserved.
3      deactivated
        The lease is deactivated, which means that it should
        not be used. Any client which is using a deactivated
        lease will be NAK'ed on their next renewal.
5      failover-updated
        The server has successfully updated the failover partner
        regarding this lease.
7      dynamic
        Last written by server which knew only about the lease
        because it was created by a send-reservation request.
13     client-reserved
        Indicates that lease is in-range on a scope that has
        restrict-to-reservations enabled.
15     use-bndack-time
        Use the time that we sent to the BNDACK as the time of lease
        expiration, so that a renewal at 1/2 the MCLT isn't required.
        Used for permanent leases and for GetNextAddress leases.
        This is persisted, but only lasts until the next BNDACK, when
        it is cleared.
```

#### fwd-dns-update-config-name [nameref](#)(0)

Names the Dns update configuration object used to perform dynamic DNS update on a forward zone.

#### giaddr [ipaddr](#)

If present, the contents of the last received non-zero giaddr field. This represents the relay agent through which the client and server last communicated.

#### lease-rebinding-time [date](#)

Displays the earliest time the client is expected to issue a rebinding request.

#### lease-renewal-time [date](#)

Displays the earliest time the client is expected to issue a renewal request.

#### limitation-id [blob](#)

Displays the value set for a client-class or client limiting the number of simultaneous active leases a DHCP server can give out to devices on customer premises.

#### parameter-request-list [option](#)

Identifies the parameter request list received from DHCP client in request packets.

#### relay-agent-auth [blob](#)

The contents of the 'authentication' suboption 8 of the relay-agent information option 82 from this client.

#### relay-agent-circuit-id [blob](#)

Displays the circuit-id sub-option of the DHCP relay-agent

information option 82 from this client.

#### **relay-agent-device-class** [int](#)

The contents of the 'device-class' suboption 4 of the relay-agent information option 82 from this client.

#### **relay-agent-option** [option](#)

Displays the contents of the relay-agent information option 82 from the most recent client interaction.

#### **relay-agent-radius-class** [string](#)

Displays the contents, if any, of the RADIUS class attribute that was contained in the RADIUS Attributes suboption of the DHCP relay-agent information option 82 from this client.

#### **relay-agent-radius-options** [blob](#)

The contents of the 'radius' suboption 7 of the relay-agent information option 82 from this client. This suboption has additional structure that is available on other attributes of this class.

#### **relay-agent-radius-pool-name** [string](#)

Displays the contents, if any, of the RADIUS framed-pool attribute 88 contained in the RADIUS attributes suboption of the DHCP relay-agent information option 82 from this client.

#### **relay-agent-radius-session-timeout** [int](#)

If present, the contents of the RADIUS 'session-timeout' attribute 27 that was contained in the RADIUS Attributes suboption 7 of the relay-agent information option 82 from this client.

#### **relay-agent-radius-user** [string](#)

Displays the contents, if any, of the RADIUS user attribute contained in the RADIUS attributes suboption of the DHCP relay-agent information option 82 from this client.

#### **relay-agent-radius-v6-pool-name** [string](#)

If present, the contents of the RADIUS 'v6-pool-name' attribute 100 that was contained in the RADIUS Attributes suboption 7 of the relay-agent information option 82 from this client.

#### **relay-agent-radius-vendor-specific** [blob](#)

If present, the contents of the RADIUS 'vendor-specific' attribute 26 that was contained in the RADIUS Attributes suboption 7 of the relay-agent information option 82 from this client.

#### **relay-agent-remote-id** [blob](#)

Displays the remote-id sub-option of the DHCP relay-agent information option 82 from this client.

#### **relay-agent-server-id-override** [ipaddr](#)

Displays the IP address in the server-id-override sub-option of the DHCP relay-agent information option 82 from this client. This value corresponds to one of two suboption numbers: If the IANA assigned suboption 152 is present in the packet, that value is returned; otherwise, if the Cisco suboption 182 is present, that value is returned.

#### **relay-agent-subnet-selection** [ipaddr](#)

Displays the IP address in the subnet selection sub-option of the DHCP relay-agent information option 82 from this client. This value corresponds to one of two suboption numbers: If the IANA assigned suboption is present in the packet, that value is returned; otherwise, if Cisco suboption 150 is present, that value is returned.

#### **relay-agent-subscriber-id** [string](#)

Displays the contents of the subscriber-id suboption of the relay-agent information option 82 from this client.

#### **relay-agent-v-i-vendor-class** [blob](#)

The contents of the 'v-i-vendor-class' suboption 9 of the relay-agent information option 82 from this client.

#### **relay-agent-vpn-id** [blob](#)

Displays the contents of the vpn-id sub-option of the DHCP relay-agent information option 82 from this client. This value corresponds to one of two suboption numbers: If the IANA assigned suboption 151 is present in the packet, that value is returned; otherwise, if Cisco suboption 181 is present, that value is returned.

**reservation-lookup-key** [blob](#)

Specifies the lookup key of the lease reservation for this lease.

**reservation-lookup-key-type** [int](#)

Specifies the data type of the reservation-lookup-key attribute. The type can be 7 for AT\_BLOB, 9 for AT\_MACADDR, or 46 for AT\_NSTRING.

**reservation-relay-agent-option** [option](#)

Displays the contents of the relay-agent information option 82 configured on a reservation for this IP address. This will be used when responding to a DHCPLEASEQUERY for this IP address in the absence of relay-agent information stored from a client interaction.

**rev-dns-update-config-name** [nameref\(0\)](#)

Names the Dns update configuration object used to perform dynamic DNS update on a reverse zone.

**scope-name** [nameref\(0\)](#)

A reference to the scope that contains this lease.

**start-time-of-state** [date](#)

Displays the time the state changed to its current value.

**state** [enumint](#)(available=1, offered=2, leased=3, expired=4, unavailable=5, released=6, other-available=7, pending-available=8)

Displays the current state of the lease.

- 1 available  
The lease is not currently leased by any client. Any client information is from the most recent client to lease or be offered this lease.
- 2 offered  
The lease is offered to the associated client. In many cases, the database is not written with information concerning offering a lease to a client since there is no requirement to update stable storage with this information.
- 3 leased  
The lease is currently leased to the client whose information appears in the lease.
- 4 expired  
The client specified has not renewed the lease, and it expired. Upon expiration the DNS information for this client was scheduled for removal.
- 5 unavailable  
The lease is unavailable. It was made unavailable because of some conflict. A ping attempt might have shown that another client using the, or the DHCP server might have detected another DHCP server handing out this IP address, or a DHCP client might have declined the lease. Use start-time-of-state to determine when the lease became unavailable, and look in the log file around that time to determine exactly why the lease became unavailable.
- 6 released  
The client specified has released the lease, but the server was configured to apply a 'release-grace-period'. The lease won't be made available until the grace-period expires.
- 7 other-available  
Used only when failover is enabled. A lease in the other-available state is available for allocation by the other server in the failover pair, but not available for allocation by this server.
- 8 pending-available  
Used only when failover is enabled. A lease in the pending-available state will be available as soon as this server can synchronize its available state with the other server.

**state-expiration-time** [date](#)

Identifies the time that we believe that the client's lease period, grace period, or release grace period will expire. A failover partner cannot, on its own, act on this time until the failover-expiration-time has also expired.

**tenant-id** [short](#) default = 0

Identifies the tenant owner of this object.

**user-defined-data** [string](#)

This string value is associated with the lease in order to allow customer applications to relate the lease record to other databases. It is not used directly by the DHCP server, but may be read and written by extensions and expressions.

**vendor-class-id** [string](#)

Displays the vendor-class-id as offered in a DHCP request option 60.

**vendor-specific-information** [blob](#)

Identifies the vendor specific information received from DHCP client in request packets.



## lease-notification

lease-notification - Reports scopes with few free leases

### Synopsis

```
lease-notification available=<number>|<percentage>
[config=<config file>]
[leasing-only]
[scopes=<scope name>|<address range>
[,<scope name>|<address range>,...]]
[[recipients=<recipient>[,<recipient>,...]]
[mail-host=<name> [errors-to=<recipient>]] ]
[vpn=<vpn-name>]
```

### Description

Use the lease-notification command to receive notification about the number of available addresses in a scope. This command reports on the scopes for which the number of available addresses falls below or equals a set value. You can specify the notification limit either as the number of free addresses or the percentage of free addresses. You can also specify who should receive e-mail notification.

Although you can use the lease-notification command interactively, its primary use is as an automated command.

You can specify clusters in several ways:

- The default cluster (localhost)
- The -C flag on the command line
- The clusters property in the config file lets you specify a group of clusters. For example, to specify several clusters, enter the following in the config file:

```
# Cluster information for summary reports
[lease-notification]
# Clustername Username Password
clusters=host1 admin passwd1, host2 admin2,host3,
host4 admin4 passwd4
```

Follow these guidelines for specifying clusters:

- Separate cluster specifications from each other with commas.
- Separate arguments for a particular cluster by whitespace.
- For long lines use continuation lines; you do not continuation escape indicators.
- Optionally, specify a user name and password for the cluster. If you do not provide a user name or password for a particular cluster, Network Registrar uses the last user name or password listed. If you do not provide user names or passwords, Network Registrar uses the information from the command line -N and -P arguments. If Network Registrar cannot find a user name or password, or the supplied user name and password are incorrect, the lease-notification command issues a warning for that cluster.

The lease-notification command output consists of an explanatory header, a table containing a row for each scope in which the number of free addresses is equal to or less than the threshold, and possible warnings related to the scopes and clusters requested.

The lease-notification command reports the following information for each scope in the table:

- Cluster name
- Scope name
- Scope network address in the canonical dotted format
- Number of high-order bits in the scope subnet mask
- Number of addresses in the scope ranges
- Percentage of addresses available for lease
- Number of addresses available for lease (addresses that are reserved or deactivated are not included as free)

## Lease Notification Keywords

### available

Specifies either a number or percentage of available addresses. If the number or percentage of available addresses is equal to or less than the specified value for the scopes being checked, Network Registrar generates a report listing information about the scopes that reach or exceed the available value.

### config

Specifies a configuration file. If you don't specify a configuration file, Network Registrar searches for the default `.nrconfig` file.

### errors-to

If you specify a mail-host, you may also specify the email address of the sender of the email in order to provide a return path for bounced email. The default value is "postmaster".

### leasing-only

Specifies that only scopes that can currently offer leases are reported.

### mail-host

The mail-host is generally already configured for the sendmail program. You can verify that your system is properly configured by issuing the command "date | mail <your-email-address>" and observing whether or not the date is emailed to you.

### recipients

If you specify the email addresses of one or more recipients, Network Registrar sends an email report to those addresses. Otherwise, Network Registrar directs the report to standard output.

### scopes

The scopes to check either by name or as a range or ranges of addresses. Network Registrar checks any scope containing any address that falls within a range of address. If you don't list any scopes or addresses, Network Registrar checks all scopes managed by the specified cluster.

### vpn

The VPN from which to select scopes to examine when executing this command. If no VPN name is specified, then the current VPN of the session is used. If the reserved VPN name "global" is used, then the global (or unnamed) VPN is used. If the reserved VPN name "all" is used, then all scopes from all vpns are examined.

## Examples

## Status

## See Also

[report](#), [export](#) addresses, [session](#) current-vpn

## lease6

lease6 - Manage DHCP lease6 objects

## Synopsis

```
lease6 list [-duid=<client-id>]
            [-lookup-key=<lookup-key> [-blob|-string]]
            [-reservation-lookup-key=<lookup-key> [-blob|-string]]
            [-cm-macaddr=<mac-addr>]
```

```

[-vpn=<vpn-name>] [-count-only]
lease6 listbrief [-duid=<client-id>]
[-lookup-key=<lookup-key> [-blob|-string]]
[-reservation-lookup-key=<lookup-key> [-blob|-string]]
[-cm-macaddr=<mac-addr>]
[-vpn=<vpn-name>] [-count-only]
lease6 [<vpn-name>/]<ipv6-address> [show]
lease6 [<vpn-name>/]<ipv6-address> get <attribute>
lease6 [<vpn-name>/]<ipv6-address> activate
lease6 [<vpn-name>/]<ipv6-address> deactivate
lease6 [<vpn-name>/]<ipv6-address> force-available
lease6 [<vpn-name>/]<ipv6-address> reconfigure
[renew|rebind|information-request] [-unicast|-via-relay]

```

## Description

The lease6 command lets you view and manipulate the current DHCPv6 leases in the cluster.

When you specify the lease on which one of these commands is to operate, you may optionally specify a <vpn-name> in which the <ipv6-address> is to be found. Specify the name of a currently defined vpn as the <vpn-name>; or use the reserved vpn name "global" (without the quotation marks) to specify the operation on leases which are not in any explicitly defined VPN. If you do not specify a vpn-name, the current VPN of the session is used.

```

lease6 list [-duid=<client-id>]
[-lookup-key=<lookup-key> [-blob|-string]]
[-macaddr=<mac-addr>] [-cm-macaddr=<mac-addr>]
[-vpn=<vpn-name>] [-count-only]

```

The list command lists DHCPv6 leases in the DHCP server. Only the leases in the current VPN or specified vpn-name are listed. The vpn-name may be "all" (without the quotation marks) to request leases in all VPNs. If -count-only is specified, only the count of the number of leases is returned (no leases are displayed). If a filter (-duid, -lookup-key, -macaddr, or -cm-macaddr) is specified, only the leases matching the filter are displayed.

```

lease6 [<vpn-name>/]<ipv6-address> activate
lease6 [<vpn-name>/]<ipv6-address> deactivate

```

The activate and deactivate commands tell the DHCP server to make the specified lease active or inactive. An inactive lease is not given out, even if it is in the available state. Making a currently leased lease inactive will not affect its behavior until it has expired and become available again.

```

lease6 [<vpn-name>/]<ipv6-address> force-available

```

The force-available command forces the specified lease into the available state.

```

lease6 [<vpn-name>/]<ipv6-address> reconfigure

```

The reconfigure command initiates sending the client a Reconfigure message (if the client and server negotiated to allow reconfigure).

## Examples

## Status

## See Also

[session](#)

## Attributes

**binding-end-time** [date](#)

Within the lease database, this holds the time when a lease binding ended.

**binding-flags** [flags](#)(virtual-binding=9)

Displays any of the following values associated with the binding:  
9 virtual-binding  
Indicates if a binding is created for GetNextAddress6 request.

**binding-iaid** [int](#)

The IAID of the binding.

**binding-rebinding-time** [date](#)

Displays the earliest time when the server requested the client to issue a Rebind request for the binding.

**binding-renewal-time** [date](#)

Displays the earliest time when the server requested the client to issue a Renew request for the binding.

**binding-start-time** [date](#)

Within the lease database, holds the time when a lease binding began.

**binding-type** [enumint](#)(IA\_NA=3, IA\_TA=4, IA\_PD=25)

Specifies the type of binding for the lease. The type number matches the DHCPv6 option number.

**client-active-leases** [int](#)

Shows the number of active leases that a client currently has in use.

**client-class-name** [nameref](#)(0)

Displays the most recently derived class name for the client.

**client-flags** [flags](#)(client-valid=1, limit-retention=5, reconfigure-active=6, binding-update-needed=8, binding-update-in-flight=9, renewal-tracked=11)

Displays any of the following values associated with the client:  
1 client-valid  
This should always be set.  
5 limit-retention  
Indicates if client's leases are subject to lease time retention restrictions.  
6 reconfigure-active  
Indicates if the client is being sent Reconfigure messages.  
8 binding-update-needed  
Indicates if a binding update is needed for the client.  
9 binding-update-in-flight  
Indicates if a binding update has been sent or is waiting to be sent for the client.  
11 renewal-tracked  
Indicates if the client's renewal is being tracked (only if DHCP server's distribute-renewal feature is enabled).

**client-id** [blob](#)

Displays the DUID of the client for the lease.

**client-last-transaction-time** [date](#)

Displays the time of last client transaction related to this lease.

**client-lookup-key** [blob](#)

Provides the lookup key for the client - it is either the client identifier (DUID) or the v6-override-client-id expression.

**client-lookup-key-type** [int](#)

Provides the data type of the client-lookup-key attribute. The type can be 7 for AT\_BLOB or 46 for AT\_NSTRING.

**client-oro** [option6](#)

Identifies the oro received from DHCP client in request messages.

**client-reconfigure-key** [blob](#)

The 128-bit key required for Reconfigure Messages to the client per the RFC 3315 Reconfigure Key Authentication Protocol.

**client-reconfigure-key-generation-time** [date](#)

The time at which the client-reconfigure-key was generated.

**client-relay-address** [ip6addr](#)

If present, displays the source address from the most recently received Relay-Forw message. If not present, the client communicated directly to the server.

#### client-relay-message [msg6](#)

Displays the most recently received relayed message. This data includes the complete Relay-Forw message(s) but excludes the client's message.

#### client-relay-port [int](#)

If present, displays the source port from the most recently received Relay-Forw message. If not present, relay used the default port (547). This is to support the "Generalized UDP Source Port for DHCP Relay" RFC.

#### client-user-defined-data [string](#)

Enables customer applications to relate the client record to other databases. It is not used directly by the DHCP server, but may be read and written by extensions and expressions.

#### client-vendor-class [nlist\(obj\(0\)\)](#)

Displays the most recently received client vendor class data. Each group of data bytes starts with the 4-byte enterprise-number followed by the vendor-class-data bytes (if any).

#### client-vendor-info [nlist\(obj\(0\)\)](#)

The most recently received vendor-specific information options data from the client. Each group of data bytes starts with the 4-byte enterprise-number followed by the option-data bytes (if any).

#### cm-mac-address [macaddr](#)

Specifies the cable modem MAC address for this lease, if applicable.

#### creation-time [date](#)

Sets the time when the lease was created.

**data-source** [enumint](#)(unknown=0, main-main=4, backup-main=5, main-backup=6, backup-backup=7, main-main-active=20, backup-main-active=21, main-backup-active=22, backup-backup-active=23, main-main-history=28, backup-main-history=29, main-backup-history=30, backup-backup-history=31)

Records the original source of the lease data and the machine from which the data was retrieved.

```
0 unknown
4 main-main
20 main-main-active
28 main-main-history
Indicates the data originated on the main server and
was retrieved from the main server.
5 backup-main
21 backup-main-active
29 backup-main-history
Indicates the data originated on the backup server and
was retrieved from the main server.
6 main-backup
22 main-backup-active
30 main-backup-history
Indicates the data originated on the main server and
was retrieved from the backup server.
7 backup-backup
23 backup-backup-active
31 backup-backup-history
Indicates the data originated on the backup server and
was retrieved from the backup.
```

The suffix -active denotes the data was returned from the active portion of the lease-state database while -history indicates that the data was from the history portion of the lease-state database. When viewing leases with the UIs, you will see all four values routinely, especially if load-balancing is enabled. When looking at lease history records, main-main and backup-backup are the usual values, but in cases where the lease history poller has determined that some data may be missing, then main-backup and backup-main can appear as well.

**dns-update-flags** [flags](#)(forward-uptodate=1, reverse-uptodate=2, update-pending=3, add-pending=4, delete-pending=5, synthesized-name=6, using-requested-fqdn=7)

The dns update flags maintained for the lease / fqdn binding.

#### excluded-prefix [ip6](#)

Specifies the excluded prefix for a prefix delegation lease if one was configured at the time the lease was assigned to the client. See RFC 6603.

#### failover-expiration-time [date](#)

Specifies the lifetime that this server has acked to the failover partner. Usually, this server must wait for this time to have expired before it can act on the state-expiration-time.

**flags** [flags](#)(reserved=1, deactivated=3, failover-updated=5, previously-reserved=8, client-reserved=13, revoked=14, use-bndack-time=15)

Flags for the lease:

```
1 (lease) reserved
The lease is reserved because of a lease reservation6 object.
The reservation-lookup-key specifies the client for which this
lease is reserved.
3 deactivated
```

The lease is deactivated, which means that it should not be used. Any client which is using a deactivated lease will be sent lifetimes of 0 on the next renewal.

5 failover-updated  
The lease is up-to-date with respect to failover (note that this is only true if the current lease's failover sequence number matches the failover object's sequence number).

8 previously-reserved  
The lease was reserved at an earlier time but is no longer. However, communication with the failover partner is needed to completely remove this reservation.

13 client-reserved  
Indicates that lease is in-range on a prefix that has restrict-to-reservations enabled.

14 revoked  
The lease is no longer usable by the client, but the client may still be using it.

15 use-bndack-time  
Use the time that we sent to the BNDACK as the time of lease expiration, so that a renewal at 1/2 the MCLT isn't required. Used for permanent leases and for GetNextAddress6 leases. This is persisted, but only lasts until the next BNDACK, when it is cleared.

**forward-dnsupdate** [nameref\(0\)](#)

Names the forward zone's DNS Update Configuration object for the lease.

**fqdn** [dname](#)

The fully qualified domain name assigned to the lease by the server (and possibly successfully entered into DNS).

**fqdn-host-label-count** [int](#) default = 1

The number of labels in the fqdn that constitute the host name portion.

**ip6address** [ip6](#)

Specifies the IPv6 address, or its prefix, of the lease.

**name-number** [int](#)

The numeric characters in the fqdn, if any, that were added by the DHCP server to disambiguate the host name.

**preferred-lifetime** [date](#)

Sets the time at which the address or prefix that was last communicated to the client is no longer preferred.

**prefix-name** [nameref\(0\)](#)

Identifies the prefix that contains this lease.

**requested-fqdn** [rel or full fqdn](#)

The partial or fully qualified domain name most recently requested by the client for the lease.

**requested-prefix-length** [rangeint\(0-128\)](#) default = 0

Specifies the requested prefix length (such as by a client in an IAPREFIX option - see RFC 3633, section 10).

**reservation-cm-mac-address** [macaddr](#)

Specifies the cable modem MAC address configured on a reservation for this client. This will be used for responding to a LEASEQUERY when no leased lease is available for this client.

**reservation-lookup-key** [blob](#)

Specifies the lookup key of the lease reservation for this lease.

**reservation-lookup-key-type** [int](#)

Specifies the data type of the reservation-lookup-key attribute. The type can be 7 for AT\_BLOB or 46 for AT\_NSTRING.

**reverse-dnsupdate** [nameref\(0\)](#)

Names the reverse zone's DNS update configuration object for the lease.

**start-time-of-state** [date](#)

Sets the time when the state last changed to its current value.

**state** [enumint\(available=1, offered=2, leased=3, expired=4, unavailable=5, released=6, other-available=7, pending-available=8, revoked=10, pending-delete=11\)](#)

Displays the current state of the lease:

- 1 available  
The lease is not currently leased by the client.
- 2 offered  
The lease is offered to the client. In many cases, the

- database is not written with information concerning offering a lease to a client since there is no requirement to update stable storage with this information.
- 3 leased  
The lease is currently leased to the client.
  - 4 expired  
The client has not renewed the lease, and it expired and will be made available after the grace period expires.
  - 5 unavailable  
The lease is unavailable. It was made unavailable because of some conflict.
  - 6 released  
The client has released the lease, but the server was configured to apply a grace period to the lease. The lease will not be made available until the grace period expires.
  - 7 other-available  
Used only when failover is enabled. A lease in the other-available state is available for allocation by the failover partner, but not available for allocation by this server.
  - 8 pending-available  
Used only when failover is enabled. A lease in the pending-available state will be available as soon as this server can synchronize its state with the failover partner.
  - 10 revoked  
The lease is no longer usable by the client, but the client may still be using it. This is DEPRECATED and no longer used (see the revoked flag in the flags attribute).
  - 11 pending-delete  
Used only when failover is enabled. A lease in the pending-delete state will be disassociated from its client as soon as this server can synchronize its state with the failover partner.

#### state-expiration-time [date](#)

Determines the earliest time at which the current state is to expire, resulting in a state transition. Possible transitions are:  
 OFFERED to deleted (if not reserved)  
 LEASED to EXPIRED  
 EXPIRED to AVAILABLE  
 RELEASED to AVAILABLE  
 AVAILABLE to deleted (if not reserved)

#### tenant-id [short](#) default = 0

Identifies the tenant owner of this object.

#### valid-lifetime [date](#)

Sets the time at which the address or prefix that was last communicated to the client is no longer valid.

#### vpn-id [int](#) default = 0, immutable

Identifies the DHCP VPN that contains this lease.

## license

license - Views and updates license information

### Synopsis

```
license register [cdns|dns|dhcp[, ...]]
    [<regional-ip>|<regional-ipv6>]
    [<regional-port>]
    [-new-uuid]

license register cdns|dns|dhcp[, ...]
    <regional-ip>
    <regional-ipv6>
    [<regional-port>]
    [-new-uuid]

license <FLEXlm-filename> create

license <key> delete
license <key> get <attribute>
license <key> [show]

license list
license listnames
license listbrief

license showUtilization [-rescan]

license showUtilHistory [-start <start-time>] [-end <end-time>]
    [-service cdns|dns|dhcp|...|all]
```

### Description

The **license** command allows you to view, create, or delete the FLEXlm licenses on a regional cluster and to register a local cluster with a regional cluster for licensing. If '-new-uuid'

option is specified, new unique universal identifier (UUID) will be generated for local cluster before registration with regional.

The **license showUtilization** command allows you to view the number of utilized IP nodes against the RTU's (Right-to-Use). If '-rescan' option is specified on the regional, a licensing scan of the local clusters is initiated to update the licensing usage.

The **license showUtilHistory** command displays license usage history for all or selected services over time. This command is only available on regional clusters.

Use **help smart** command to get details for smart license configuration commands.

## Examples

```
nrcmd> license register dhcp 192.168.0.1 1244
nrcmd> license register cdns 2001:DB8::1 1244
nrcmd> license register dhcp,dns 192.168.0.1 2001:DB8::1 1244
```

## Status

## See Also

The "Network Registrar CLI Introduction" section describes how licenses are used in nrcmd.

## link

link - configures IPv6 network links for use in DHCPv6

Note: dhcp-link is a synonym for compatibility with earlier versions.

## Synopsis

```
link list
link listnames
link listbrief
link <name> create [[template-root-prefix=<prefix>]
                  [template=<template-name>]
                  [<attribute>=<value> ...]

link <name> delete
link <name> set <attribute>=<value> [<attribute>=<value> ...]
link <name> get <attribute>
link <name> unset <attribute>
link <name> enable <attribute>
link <name> disable <attribute>
link <name> [show]

link <name> listPrefixes
link <name> listPrefixNames

link listGroups
link listGroupNames
link listGroup <group-name>

link <name> applyTemplate <template-name> [<template-root-prefix>]

link <name> push <cluster/failover-pair-list>
               [-template=<prefix-template-name>]
               [-omitparents] [-omitchildren] [-report]

link <name> reclaim [<cluster/failover-pair-list>] [-force] [-report]
```

## Description

The link command configures IPv6 network links. Links group IPv6 prefixes (see the prefix command) together. Links are required if multiple prefixes share the same physical link.

When creating a link using a template, specify - for the <name> to



allow the link template's link-name-expr to name the link.

The listGroups, listGroupNames, and listGroup operations are helpful when using link groups to examine the configuration.

The push and reclaim commands are only available when connected to a regional cluster. For push, usually only a single cluster or failover-pair may be specified, and for reclaim no cluster or failover-pair. However, a list of clusters / failover-pairs may be specified if the link is a universal link.

## Examples

## Status

## See Also

[link-template](#), [prefix](#)

### Attributes

**description** [string](#)

Describes the link.

**embedded-policy** [obj\(0\)](#)

Refers to a policy embedded within a single specific link object used when replying to clients.

**free-address-config** [nameref\(0\)](#)

Identifies which trap captures unexpected free address events on this Link.  
If this attribute is not configured, the server looks for the v6-default-free-address-config on the DHCPSErver object.

**group-name** [string](#)

Specifies the link group to which this link belongs. When servicing client requests, the prefixes under the links in the link group may be used by clients.

**interface** [oid](#) unique

Specifies the router-interface associated with this link.

**local-cluster** [oid](#)

Identifies the local DHCP cluster or failover pair for this regional link.

**maintenance** [bool](#) default = disabled

If this attribute is enabled, all Prefixes under this link will participate in the current maintenance window. DHCP clients will have their renewal times adjusted to avoid renewals during the maintenance window. In addition, they will be directed to renew during the renewal-period after the end of the maintenance window.

**name** [string](#) required,unique

Provides user-assigned name for the link.

**owner** [nameref\(0\)](#)

Identifies the owner of this link, referenced by name. Owners can be used to limit administrative access by owner.

**policy** [nameref\(0\)](#) default = default

Refers to a shared policy used when replying to clients.

**prefix-list** [nlist\(obj\(0\)\)](#)

Lists the prefixes to be associated with the link. This attribute is used to add or modify the link and its prefixes in a single database action. All objects must be valid, or none will be accepted. The associated prefixes will be stored separately, and will not be returned in the parent link object.

**region** [nameref\(0\)](#)

Identifies the region for this link, referenced by name. Regions can be used to limit administrative access by region.

**template-root-prefix** [prefix](#)

Identifies the root prefix address for prefixes associated with the link. This attribute is used when processing a link template that defines associated prefixes.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**type** [enumint](#)(topological=0, location-independent=1, universal=2) default = topological

Specifies the type of link.  
Topological links are used to determine where in the network a client is located.  
Location-independent links must be part of a link group and are available to clients on the other links in the link group. Each link group may have only a single location-independent link.  
A universal link is available to all clients on the VPN. Each VPN may have only a single universal link. A universal link can not be part of a link group.

**vpn-id** [int](#) default = 0, immutable

Identifies the VPN that contains the link.

## link-policy

link-policy - Edits the DHCP policy embedded in a link.

Note: dhcp-link-policy is a synonym for compatibility with earlier versions.

### Synopsis

```
link-policy <name> delete
link-policy <name> set <attribute>=<value>
                        [<attribute>=<value> ...]
link-policy <name> get <attribute>
link-policy <name> disable <attribute>
link-policy <name> enable <attribute>
link-policy <name> show

link-policy <name> setV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
link-policy <name> addV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
link-policy <name> getV6Option <opt-name | id>[.<instance>]
link-policy <name> unsetV6Option <opt-name | id>[.<instance>]
link-policy <name> listV6Options

link-policy <name> setV6VendorOption <opt-name | id>
                        <opt-set-name> <value> [-blob | -expression]
link-policy <name> getV6VendorOption <opt-name | id>
                        <opt-set-name>
link-policy <name> unsetV6VendorOption <opt-name | id>
                        <opt-set-name>
link-policy <name> listV6VendorOptions
```

### Description

The link-policy command lets you configure a DHCP policy embedded in a DHCP link. An embedded policy is a collection of DHCP option values and settings that are associated with (and named by) another object -- in this case a link. You create a link-policy when you first reference it, and you delete it when you delete the link.

See the help file for the policy command for more information.

### Examples

### Status

## See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the `inhibit-all-renews` attribute or the `inhibit-renews-at-reboot` attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the `FQDN` option to indicate that it wants to do the A record update in the request, and if this value is `TRUE`, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to `FALSE`, Rapid Commit is not allowed.
- If at least one has it set to `TRUE`, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

### allow-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

### default-prefix-length [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

### excluded-prefix [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603). Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified. Using the example in RFC 6603, this attribute would be set to `0:0:0:f::/64`. If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

### forward-dnsupdate [nameref](#)(0)

Specifies the name of the update configuration that determines

which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1).  
The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server.  
The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server.  
The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only.

Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- |   |          |   |
|---|----------|---|
| 1 | allow    | Allows clients to request reconfiguration support and the server will honor the request (default).  |
| 2 | disallow | Allows clients to request reconfiguration support but the server will not honor the clients' request.   |
| 3 | require  | Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option. |

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

**reconfigure-via-relay** [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

**reverse-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### **server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### **shortest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### **split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### **tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### **unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system\_default\_policy, then the default is 86400 seconds (or 24 hours).

#### **use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### **v4-bootp-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all BOOTP clients.

#### **v4-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### **v6-max-client-preferred-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### **v6-max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### **v6-max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### **v6-max-client-valid-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it

does not impact any of the time calculations made by the server.

#### **v6-reply-options** [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### **v6-rsoo-allowed-options** [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### **valid-lifetime** [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### **view-id** [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## link-template

link-template - Configures a link template

### Synopsis

```
link-template list
link-template listnames
link-template listbrief

link-template <name> create [<attribute>=<value>...]
link-template <name> delete
link-template <name> set <attribute>=<value> [<attribute>=<value> ...]
link-template <name> get <attribute>
link-template <name> unset <attribute>

link-template <name> disable <attribute>
link-template <name> enable <attribute>
link-template <name> show

link-template <name> create clone=<clone-name>
link-template <name> apply-to all | <link1>[,...]
link-template <name> apply-to <link> [<prefix>]

link-template < <name> | all > pull < ensure | replace | exact >
                        <cluster-name> [-report-only | -report]
link-template < <name> | all > push < ensure | replace | exact >
                        <cluster-list> [-report-only | -report]
link-template <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The link-template command lets you configure a template to use when creating links.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

### Examples

## Status

## See Also

[link](#)

## Attributes

### description [string](#)

Describes the link template.

### embedded-policy [obj](#)(0)

Specifies an policy embedded.  
Note: When the template is applied, this will replace the entire embedded-policy in the link.

### free-address-config [nameref](#)(0)

Identifies which trap captures unexpected free address events on this link.  
If this attribute is not configured, the server looks for the v6-default-free-address-config on the DHCPserver object.

### group-name [string](#)

Specifies the link group to which this link belongs. When servicing client requests, the prefixes under the links in the link group may be used by clients.

### link-description-expr [expr](#)

An expression to define the description on the link object created when using the template.

### link-name-expr [expr](#)

An expression to define the name of the link object created when using the template.

### name [string](#) required,unique

Assigns a name to this link template.

### options-expr [expr](#)

An expression to define the list of embedded policy options to be created.

### owner [nameref](#)(0)

Identifies the owner of this link, referenced by name. Owners can be used to limit administrative access by owner.

### policy [nameref](#)(0) default = default

Refers to a shared policy used when replying to clients.

### prefix-expr [expr](#)

Defines an expression used to create the list of associated prefixes.

### region [nameref](#)(0)

Identifies the region for this link, referenced by name. Regions can be used to limit administrative access by region.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

### type [enumint](#)(topological=0, location-independent=1, universal=2) default = topological

Specifies the type of link.  
Topological links are used to determine where in the network a client is located.  
Location-independent links must be part of a link group and are available to clients on the other links in the link group. Each link group may have only a single location-independent link.  
A universal link is available to all clients on the VPN. Each VPN may have only a single universal link. A universal link can not be part of a link group.



link-template-policy - Edits the DHCP policy embedded in a link-template

## Synopsis

```
link-template-policy <name> delete
link-template-policy <name> set
<attribute>=<value> [...]
link-template-policy <name> get <attribute>
link-template-policy <name> disable <attribute>
link-template-policy <name> enable <attribute>
link-template-policy <name> show

link-template-policy <name> setV6Option
<opt-name | id>[.<instance>] <value>
[-blob | -expression] [-roundrobin]
link-template-policy <name> addV6Option
<opt-name | id>[.<instance>] <value>
[-blob | -expression] [-roundrobin]
link-template-policy <name> getV6Option <opt-name | id>[.<instance>]
link-template-policy <name> unsetV6Option <opt-name | id>[.<instance>]
link-template-policy <name> listV6Options

link-template-policy <name> setV6VendorOption <opt-name | id>
<opt-set-name> <value> [-blob | -expression]
link-template-policy <name> getV6VendorOption <opt-name | id>
<opt-set-name>
link-template-policy <name> unsetV6VendorOption <opt-name | id>
<opt-set-name>
link-template-policy <name> listV6VendorOptions
```

## Description

The link-template-policy command lets you configure a DHCP policy embedded in a DHCP link template. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case a link template. A link-template-policy is created implicitly when you first reference it, and is deleted when the link-template is deleted.

See the help file for the policy command for more information.

## Examples

```
nrncmd> link-template-policy examplelink set forward-zone-name=example.com.
nrncmd> link-template-policy examplelink enable allow-rapid-commit
```

## Status

## See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the inhibit-all-renews attribute or the inhibit-renews-at-reboot attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

**allow-dual-zone-dns-update** [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

**allow-lease-time-override** [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

**allow-non-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

**allow-rapid-commit** [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603). Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified. Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64. If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object. Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCPACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action). This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet. Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebind operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer

than 64 characters.

#### packet-siaddr [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

#### permanent-leases [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

#### preferred-lifetime [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted.  
Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

#### reconfigure [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- 1 allow Allows clients to request reconfiguration support and the server will honor the request (default).
- 2 disallow Allows clients to request reconfiguration support but the server will not honor the clients' request.
- 3 require Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option.

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

#### reconfigure-via-relay [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

#### reverse-dnsupdate [nameref](#)(0)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### server-lease-time [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### shortest-prefix-length [rangeint](#)(0-128)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### split-lease-times [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system\_default\_policy, then the default is 86400 seconds (or 24 hours).

**use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

**v4-bootp-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

**v4-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

**v6-max-client-preferred-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

**v6-max-client-rebinding-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

**v6-max-client-renewal-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

**v6-max-client-valid-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

**v6-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

**v6-rs00-allowed-options** [nlist\(obj\(0\)\)](#)

Lists the RS00-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RS00 (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

**valid-lifetime** [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this

period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime.  
Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## no

no - Unset specified configuration

### Synopsis

```
no license smart enable
no license smart reservation
no license smart custom_id
no license smart transport
no license smart url
no license smart http-proxy
no license smart privacy <all|hostname|version>
no debug smart_lic <error|trace|debug|all>

no call-home destination address http
no call-home http-proxy
no call-home data-privacy hostname
no call-home debug <error|trace|detail|all>
```

### Description

These commands are available in smart license configuration mode.

The **no license smart** command allows you to unset the specified smart license configuration.

The **no call-home** command allows you to unset the specified call home configuration.

The **no debug smart\_lic** command allows you to unset the smart license debug level.

### Examples

### Status

### See Also

[smart](#), [call-home](#)

## option

option - Configures option definitions

### Synopsis

```
option <id> <option-setname> create <option-name> <type> [<attribute>=<value>]
option <name | id> <option-setname> delete
option <name | id> <option-setname> set <attribute>=<value> [<attribute>=<value>...]
option <name | id> <option-setname> get <attribute>
option <name | id> <option-setname> [show]
option <name | id> <option-setname> unset <attribute>

option <name | id> <option-setname> enable <attribute>
option <name | id> <option-setname> disable <attribute>

option <option-setname> list
```

**option** <option-setname> **listnames**  
**option** <option-setname> **show**

**option listtypes**

## Description

The option command configures option definitions.

Use reserved names as follows:

dhcp-config and dhcp6-config to view currently configured option sets for DHCPv4 and DHCPv6 respectively.

dhcp-custom and dhcp6-custom to view/add/modify/delete custom option definitions.

NOTE: you may also use dhcp-config and dhcp6-config to add/modify/delete custom option definitions. These names are used to operate on the respective custom set.

Changes to the custom sets are merged with the built-in option definitions to form the config sets.

Modifications to the custom sets are not visible in the config sets until after a SAVE is performed.

You cannot use the reserved names when creating or deleting an option set. The custom sets are created when the first custom option definition is created.

Use the listtypes command to view the list of option types available for use in creating custom option definitions.

## Examples

## Status

## See Also

### Attributes

#### expression [expr](#)

The expression which is evaluated to provide the option data.  
Note: That this is only used if there is no value attribute.

#### number [int](#)

The option number (the T part of the TLV)

#### option-definition-set-name [nameref\(0\)](#)

A reference to the option-definition-set used to create this option instance.

#### option-desc [objref\(0\)](#)

The option description that provides the type and name for this specific option. This is a shortcut for a lookup by id in the appropriate TLV desc table.

#### round-robin [bool](#) default = false

Determines whether round robin ordering should be enabled for this option. When enabled and option data is list of values, the list order will be rotated for each client based on hash of client's identifier. Thus, a client will always get same order.

#### sub-options [nlist\(obj\(0\)\)](#)

This attribute provides for the optional subdivision of option data into nested sub-option objects. This is not used for storing DHCP server configuration, but may be useful in other uses of TLV based values.

value [blob](#)

The option value (the V part of the TLV)

## option-set

option-set - Configure option definition sets

### Synopsis

```
option-set list
option-set listnames

option-set <name> create <8-bit | 16-bit>
                    vendor-option-string=<string>
                    [vendor-option-regex=<string>]
                    [<attribute>=<value>]
option-set <name> create <8-bit | 16-bit>
                    vendor-option-enterprise-id=<integer>
                    [<attribute>=<value>]
option-set <name> delete

option-set <name> [show]
option-set <name> list
option-set <name> listnames
option-set <name> get <attribute>

option-set <name> set <attribute>=<value>
                    [<attribute>=<value> ...]
option-set <name> unset <attribute>
option-set <name> enable <attribute>
option-set <name> disable <attribute>

option-set < <name> | all > pull < ensure | replace | exact >
                    <cluster-list> [-report-only | -report]
option-set < <name> | all > push < ensure | replace | exact >
                    <cluster-list> [-report-only | -report]
option-set <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The option-set command configures option definition sets.

Use reserved names as follows:

dhcp-config and dhcp6-config to view currently configured options for DHCPv4 and DHCPv6 respectively. These reserved names show all built-in and custom option definitions. These option sets can not be created nor deleted.

dhcp-custom and dhcp6-custom to view only the custom option definitions. These names can be used on a delete to delete all custom definitions.

Changes to the custom sets are merged with the built-in option definitions to form the config sets.

Modifications to the custom sets are not visible in the config sets until after you perform a SAVE.

Use 8-bit to create a dhcpv4 vendor option definition set.  
Use 16-bit to create a dhcpv6 vendor option definition set.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

### Examples

### Status

### See Also



## [export option-set](#)

### Attributes

desc [string](#)

Describes this option definition set.

flags [flags](#)(builtin=1, customized=2, vendor=3) transient

Reports the flags for this set of options.

id-range [enumint](#)(one-byte=1, two-bytes=2, four-bytes=4) required

Specifies the byte size of the type identifiers for all options in this set. Values can be:

- 1 - one byte
- 2 - two bytes
- 4 - four bytes

name [string](#) required,unique

Designates the name of this option definition set.

tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

vendor-option-enterprise-id [int](#)

The enterprise identifier.  
If defined, contains definitions for any option with AT\_VENDOR\_OPTS or AT\_VENDOR\_CLASS type, such as

- DHCPv6 id 16
- DHCPv6 id 17
- DHCPv4 id 124
- DHCPv4 id 125

This attribute is required for a DHCPv6 vendor option.  
This attribute optionally can be set in combination with the vendor-option-string for a DHCPv4 vendor option.

vendor-option-regex [string](#)

Contains the vendor option regex string to match with vendor class identifier string provided by the DHCP client device vendor in DHCPv4 option-60. The DHCP server uses this string to identify the DHCP client device and return appropriate vendor specific option data.  
The server ignores this attribute for DHCPv6.

vendor-option-string [string](#)

Contains the vendor class identifier string provided by the DHCP client device vendor in DHCPv4 option-60. The DHCP server uses this string to identify the DHCP client device and return appropriate vendor specific option data.  
The server ignores this attribute for DHCPv6.

## owner

owner - Configures owners

### Synopsis

```
owner <tag> create <name> [<attribute>=<value>]
owner <tag> delete
owner list
owner listnames
owner listbrief
owner <tag> show
owner <tag> set <attribute>=<value> [<attribute>=<value> ...]
owner <tag> get <attribute>
owner <tag> unset <attribute>

owner <tag> enable <attribute>
owner <tag> disable <attribute>

owner < <tag> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
owner < <tag> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
owner <tag> reclaim <cluster-list> [-report-only | -report]
```

### Description

The owner command configures a specified owner. You create owners and associate them with address blocks, subnets, and zones.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

## Status

## See Also

### Attributes

**contact** [string](#)

Provides contact information for this owner.

**name** [string](#)

Displays the full name, or printable name, for this owner.

**organization** [nameref\(0\)](#)

Specifies the organization name required for ARIN reporting purposes.

**tag** [string](#) required,unique

Displays a unique tag name for this owner. Typically, it is a short name referring to this owner.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## prefix

prefix - Configures IPv6 network prefixes for use in DHCPv6

Note: dhcp-prefix is a synonym for compatibility with earlier versions of Network Registrar.

## Synopsis

```
prefix list
prefix listnames
prefix listbrief
prefix <name> create <address> [template=<template-name>]
                        [<attribute>=<value>]
prefix <name> delete
prefix <name> set <attribute>=<value> [<attribute>=<value> ...]
prefix <name> get <attribute>
prefix <name> unset <attribute>
prefix <name> enable <attribute>
prefix <name> disable <attribute>
prefix <name> [show]

prefix <name> listLeases

prefix <name> addReservation <address> <lookup key>
prefix <name> removeReservation <address>
prefix <name> listReservations

prefix <name> applyTemplate <template-name>

prefix <name> getUtilization

prefix <name> push <cluster/failover-pair-list> [-template=<template-name>]
                [-omitparents] [-omitchildren] [-report]
prefix <name> reclaim [<cluster/failover-pair-list>] [-force]
                [-omitchildren] [-report-only | -report]
```

## Description

The prefix command configures IPv6 network prefixes. Prefixes configure DHCPv6 address allocation and prefix delegation.

When creating a prefix using a template, specify - for the <name> to allow the prefix template's prefix-name-expr to name the prefix.

The push and reclaim commands are only available when connected to a regional cluster. For push, usually only a single cluster or failover-pair may be specified, and for reclaim no cluster or failover-pair. However, a list of clusters / failover-pairs may be specified if the prefix is on a universal link.

## Examples

```
nrcmd> prefix example-pref create ff00::/8
nrcmd> prefix example-pref set address=ff00::/10
```

## Status

## See Also

[link](#), [prefix-template](#)

## Attributes

address [prefix](#) required,immutable

Identifies a prefix (subnet) that an interface belongs to using the high-order bits of an IPv6 address.

allocation-algorithms [flags](#)(client-request=1, reservation=2, extension=3, interface-identifier=4, random=5, best-fit=6) default = reservation,extension,random,best-fit

Controls the algorithms used by the server to select a new address or prefix to lease to a client. Note however that when the prefix's restrict-to-reservations is enabled, only the reservation flag is used (the others are ignored as only lease or client reservations can be used).

The available algorithms are:

client-request

This setting (off by default) controls whether the server uses a client requested lease.

reservation

This setting (on by default) controls whether the server uses an available lease reservation for the client.

extension

This setting (on by default) controls whether the server calls extensions attached at the generate-lease extension point to generate an address or prefix for the client.

interface-identifier

This setting (off by default) controls whether the server uses the interface-identifier from the client's (link-local) address to generate an address. It is ignored for temporary addresses and prefix delegation.

random

This setting (on by default) controls whether the server generates an address using an RFC 3041-like algorithm. It is ignored for prefix delegation.

best-fit

This setting (on by default) controls whether the server will delegate the first, best-fit available prefix. It is ignored for addresses.

When the server needs an address to assign to a client, it processes the flags in the following order: client-request, reservation, extension, interface-identifier, and random. Processing stops when a usable address is produced.

When the server needs to delegate a prefix to a client, it processes the flags in the following order: client-request, reservation, extension, best-fit. Processing stops when a usable prefix is produced.

allocation-group [string](#)

Specifies the allocation group to which this prefix belongs. At most one lease per binding is allocated across all of the prefixes on a link with the same allocation group name. This name is only specific to the link. Different links can reuse the same allocation group names.

allocation-group-priority [int](#) default = 0

Specifies the priority of this prefix over other prefixes in the same allocation group. This settings allows control over which prefixes are used before others. Lower numeric values have higher priorities, but a priority of 0 (the default) has the lowest possible priority.

Prefixes with the same priority are dynamically ordered by active lease count, where the lowest count will have the highest priority.

**deactivated** [bool](#) default = disabled

Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients. It treats all addresses in its ranges as if they were individually deactivated.  
Default, false (active).

**description** [string](#)

Describes the prefix.

**dhcp-type** [enumint](#)(stateless=0, dhcp=1, prefix-delegation=2, infrastructure=3, parent=4) default = dhcp

Defines how DHCP manages address assignment within a prefix:

stateless	Prefix is used only for stateless option configuration.
dhcp	Prefix is used for stateful address assignment.
prefix-delegation	Prefix is used for prefix-delegation.
infrastructure	Prefix is used to map a client address to a link, and does not have an address pool.
parent	Prefix is not used by DHCP. It is used as a container object, to group child prefixes.

**embedded-policy** [obj](#)(0)

Specifies a policy embedded within an IPv6 prefix.

**expiration-time** [date](#)

Sets the time and date on which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix. Once the expiration-time has passed, the prefix is no longer used (though old leases and leases with grace or affinity periods continue to exist until those periods elapse). Enter this as a date in the format "[weekday] mon day hh:mm[:ss] year". For example, "Dec 31 23:59 2006".

**free-address-config** [nameref](#)(0)

Identifies which trap captures unexpected free address events on this prefix. If this attribute is not configured, the server looks for the free-address-config attribute on the parent Link object. If that attribute is not configured, the server looks for the v6-default-free-address-config on the DHCPServer object.

**ignore-declines** [bool](#) default = false

Controls whether the DHCP server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix. If enabled, the DHCP server ignores all declines for leases in this prefix. If disabled or not set, the DHCP server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE message if it is leased to the client. The default value is false, so that DECLINE messages are processed normally.

**link** [nameref](#)(0)

Associates an IPv6 prefix (subnet) with a link. Use this attribute to group prefixes that are on a single link.

**local-cluster** [oid](#)

Identifies the local DHCP cluster or failover pair for this regional prefix.

**maintenance** [bool](#) default = disabled

If this attribute is enabled, this Prefix and any other Prefixes on the same Link will participate in the current maintenance window. DHCP clients will have their renewal times adjusted to avoid renewals during the maintenance window. In addition, they will be directed to renew during the renewal-period after the end of the maintenance window.

**max-leases** [rangeint](#)(0-2000000) default = 65536

Sets the maximum number of non-reserved leases that the server will allow to exist on this prefix. When a new lease needs to be created, the server will only do so if the limit has not been exceeded. When the limit is exceeded, no new leases can be created and offered to clients. This limit is not applied when existing leases are loaded from the lease state database during server start-up. This attribute is also used when calculating the free-address level for address traps.

**max-pd-balancing-length** [rangeint](#)(1-128) default = 64

Specifies the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix. If specified, this value must be equal or less than the prefix range's prefix length plus 24, otherwise the server will use the

prefix's range plus 24. Thus, a /32 prefix will only perform pool balancing for prefixes up to a /56. If unspecified, the lessor of the prefix range's prefix length plus 24 or 64 is used. Available prefix-delegation prefixes longer than this value will not be considered when balancing prefix-delegation pools. This value should be the same or smaller than the longest allowed prefix length to assure all prefix-delegation prefixes can be assigned to clients as balancing prefixes longer than this length would result in prefixes that can never be delegated to clients.

**name** [string](#) required,unique

Assigns a name to an IPv6 prefix (subnet).

**owner** [nameref](#)(0)

Identifies the owner of this prefix, referenced by name. Owners can be used to limit administrative access to prefixes by owner. If the prefix has an associated link, the owner of the associated link will apply, if it is set. If there is no associated link, or the link owner is unset, the owner of the parent prefix will apply, if the prefix owner is unset.

**policy** [nameref](#)(0)

Refers to a shared policy to use when replying to clients.

**range** [prefix](#)

Specifies a prefix contained by the prefix address to limit the range of addresses or prefixes available for assignment. If unspecified on a prefix of dhcp-type 'prefix-delegation', the ranges of other prefixes with the same prefix address will be excluded from this prefix. If unspecified on a prefix of other dhcp-type values, the prefix address is used.

**range-end** [ip6](#)

Specifies the end address within the prefix address range that will be used to allocate leases, if this is a DHCP type prefix. If unset, the last available address in the prefix address range is used as the end address (except if range-start was specified as a prefix, in which case the last address of the prefix specified by range-start is used).

**range-start** [ip6](#)

Specifies either the start address within the prefix address range that will be used to allocate leases or a prefix which will be used as the range (range-end must not be specified in this case), if this is a DHCP type prefix. If unset, the first available address in the prefix address range is used as the start address.

**region** [nameref](#)(0)

Identifies the region for this prefix, referenced by name. Regions can be used to limit administrative access to prefixes by region. If the prefix has an associated link, the region for the associated link will apply, if it is set. If there is no associated link, or the link region is unset, the region for the parent prefix will apply, if the prefix region is unset.

**restrict-to-admin-allocation** [bool](#) default = disabled

Controls whether the prefix is restricted to administrative requests to allocate the next available address. If set, the server will only respond to a client with an address from this prefix if it has been pre-allocated to the client.

**restrict-to-reservations** [bool](#) default = disabled

Controls whether the prefix is restricted to client (or lease) reservations. If enabled, the DHCP server will not automatically assign addresses or delegate prefixes to clients but instead requires the address or prefix to be supplied by a reservation, either a lease reservation or a client reservation, which is specified via a client entry or through an extension and the environment dictionary.

**reverse-zone-prefix-length** [rangeint](#)(0-124)

Specifies the prefix length of the reverse zone for ip6.arpa updates. You do not need to specify the full reverse zone, because you can synthesize it by using the ip6.arpa domain. Use a multiple of 4 for the value, because ip6.arpa zones are on 4-bit boundaries. If not a multiple of 4, the value is rounded up to the next multiple of 4. The maximum value is 124, because specifying 128 would create a zone name without any possible hostnames contained within. A value of 0 means none of the bits are used for the zone name, hence ip6.arpa is used. If you omit the value from the DNS update configuration, the server uses the value from the prefix or, as a last resort, the prefix length derived from the address value of the prefix.

**selection-tags** [nlist](#)(obj(0))

Associates selection tags with an IPv6 prefix (subnet).

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**vpn-id** [int](#) default = 0, immutable

Identifies the VPN that contains an IPv6 prefix.

## prefix-policy

prefix-policy - Edits the DHCP policy embedded in a prefix

Note: dhcp-prefix-policy is a synonym for compatibility with earlier versions of Network Registrar.

### Synopsis

```
prefix-policy <name> delete
prefix-policy <name> set <attribute>=<value>
                        [<attribute>=<value> ...]
prefix-policy <name> get <attribute>
prefix-policy <name> disable <feature>
prefix-policy <name> enable <feature>
prefix-policy <name> show

prefix-policy <name> setV6Option <opt-name | id>[.<instance>]
                                <value> [-blob | -expression] [-roundrobin]
prefix-policy <name> addV6Option <opt-name | id>[.<instance>]
                                <value> [-blob | -expression] [-roundrobin]
prefix-policy <name> getV6Option <opt-name | id>[.<instance>]
prefix-policy <name> unsetV6Option <opt-name | id>[.<instance>]
prefix-policy <name> listV6Options

prefix-policy <name> setV6VendorOption <opt-name | id> <opt-set-name>
                                <value> [-blob | -expression]
prefix-policy <name> getV6VendorOption <opt-name | id> <opt-set-name>
prefix-policy <name> unsetV6VendorOption <opt-name | id> <opt-set-name>
prefix-policy <name> listV6VendorOptions
```

### Description

The prefix-policy command lets you configure a DHCP policy that is embedded in a DHCP prefix. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case a prefix. A prefix-policy is created implicitly when you first reference it, and is deleted when the prefix is deleted.

See the help file for the policy command for more information.

### Examples

### Status

### See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-template-policy](#), [scope-policy](#), [scope-template-policy](#)

### Attributes

affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the inhibit-all-renews attribute or the inhibit-renews-at-reboot attribute.

**allow-client-a-record-update** [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

**allow-dual-zone-dns-update** [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

**allow-lease-time-override** [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

**allow-non-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA NA) addresses. The default is to allow clients to request non-temporary addresses.

**allow-rapid-commit** [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603). Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified. Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64. If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object. Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCPACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action). This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet. Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client.



The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- 1 allow Allows clients to request reconfiguration support and the server will honor the request (default).
- 2 disallow Allows clients to request reconfiguration support but the server will not honor the clients' request.
- 3 require Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option.

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

**reconfigure-via-relay** [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

**reverse-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

**server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

**shortest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system default\_policy, then the default is 86400 seconds (or 24 hours).

**use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

**v4-bootp-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

**v4-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

**v6-max-client-preferred-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

**v6-max-client-rebinding-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

**v6-max-client-renewal-time** [rangetime\(30s-2y\)](#)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

**v6-max-client-valid-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

**v6-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

**v6-rsoo-allowed-options** [nlist\(obj\(0\)\)](#)

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

`valid-lifetime` [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime.  
Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

`view-id` [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## prefix-template

`prefix-template` - Configures a prefix template.

### Synopsis

```
prefix-template list
prefix-template listnames
prefix-template listbrief

prefix-template <name> create [<attribute>=<value> ...]
prefix-template <name> delete
prefix-template <name> set <attribute>=<value> [<attribute>=<value> ...]
prefix-template <name> get <attribute>
prefix-template <name> unset <attribute>

prefix-template <name> disable <attribute>
prefix-template <name> enable <attribute>
prefix-template <name> show

prefix-template <name> create clone=<clone-name>
prefix-template <name> apply-to <all | <prefix1>[,...]>

prefix-template < <name> | all > pull < ensure | replace | exact >
  <cluster-name> [-report-only | -report]
prefix-template < <name> | all > push < ensure | replace | exact >
  <cluster-list> [-report-only | -report]
prefix-template <name> reclaim <cluster-list> [-report-only | -report]
```

### Description

The `prefix-template` command lets you configure a template to use when creating prefixes.

The `pull`, `push`, and `reclaim` commands are only available when connected to a regional cluster. For `push` and `reclaim`, a list of clusters or "all" may be specified.

### Examples

### Status

### See Also

[prefix](#)

### Attributes

`allocation-algorithms` [flags](#)(client-request=1, reservation=2, extension=3, interface-identifier=4, random=5, best-fit=6) default = reservation,extension,random,best-fit

Controls the algorithms used by the server to select a new address or prefix to lease to a client. Note however that when the prefix's `restrict-to-reservations` is enabled, only the reservation flag is used (the others are ignored as only lease or client reservations can be used).  
The available algorithms are:  
`client-request`  
This setting (off by default) controls whether the server uses a client requested lease.

reservation

This setting (on by default) controls whether the server uses an available lease reservation for the client.

extension

This setting (on by default) controls whether the server calls extensions attached at the generate-lease extension point to generate an address or prefix for the client.

interface-identifier

This setting (off by default) controls whether the server uses the interface-identifier from the client's (link-local) address to generate an address. It is ignored for temporary addresses and prefix delegation.

random

This setting (on by default) controls whether the server generates an address using an RFC 3041-like algorithm. It is ignored for prefix delegation.

best-fit

This setting (on by default) controls whether the server will delegate the first, best-fit available prefix. It is ignored for addresses.

When the server needs an address to assign to a client, it processes the flags in the following order: client-request, reservation, extension, interface-identifier, and random. Processing stops when a usable address is produced.

When the server needs to delegate a prefix to a client, it processes the flags in the following order: client-request, reservation, extension, best-fit. Processing stops when a usable prefix is produced.

#### allocation-group [string](#)

Specifies the allocation group to which this prefix belongs. At most one lease per binding is allocated across all of the prefixes on a link with the same allocation group name. This name is only specific to the link. Different links can reuse the same allocation group names.

#### allocation-group-priority [int](#) default = 0

Specifies the priority of this prefix over other prefixes in the same allocation group. This settings allows control over which prefixes are used before others. Lower numeric values have higher priorities, but a priority of 0 (the default) has the lowest possible priority. Prefixes with the same priority are dynamically ordered by active lease count, where the lowest count will have the highest priority.

#### deactivated [bool](#) default = disabled

Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients. It treats all addresses in its ranges as if they were individually deactivated. Default, false (active).

#### description [string](#)

Describes the prefix template.

#### dhcp-type [enumint](#)(stateless=0, dhcp=1, prefix-delegation=2, infrastructure=3, parent=4) default = dhcp

Defines how DHCP manages address assignment within a prefix:

stateless	Prefix is used only for stateless option configuration.
dhcp	Prefix is used for stateful address assignment.
prefix-delegation	Prefix is used for prefix-delegation.
infrastructure	Prefix is used to map a client address to a link, and does not have an address pool.
parent	Prefix is not used by DHCP. It is used as a container object, to group child prefixes.

#### embedded-policy [obj](#)(0)

Specifies an policy embedded.

Note: When the template is applied, this will replace the entire embedded-policy in the prefix.

#### expiration-time [date](#)

Sets the time and date on which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix.

Once the expiration-time has passed, the prefix is no longer used (though old leases and leases with grace or affinity periods continue to exist until those periods elapse).

Enter this as a date in the format "[weekday] mon day hh:mm[:ss] year". For example, "Dec 31 23:59 2006".

#### free-address-config [nameref](#)(0)

Identifies which trap captures unexpected free address events on this prefix.

If this attribute is not configured, the server looks for the free-address-config attribute on the parent Link object.

If that attribute is not configured, the server looks for the v6-default-free-address-config on the DHCPv6Server object.

#### ignore-declines [bool](#) default = false

Controls whether the DHCP server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix.

If enabled, the DHCP server ignores all declines for leases in this prefix.

If disabled or not set, the DHCP server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE

message if it is leased to the client.  
The default value is false, so that DECLINE messages are processed normally.

**max-leases** [rangeint](#)(0-2000000) default = 65536

Sets the maximum number of non-reserved leases that the server will allow to exist on the prefix. When a new lease needs to be created, the server will only do so if the limit has not been exceeded. When the limit is exceeded, no new leases can be created and offered to clients.

**max-pd-balancing-length** [rangeint](#)(1-128) default = 64

Specifies the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix.  
If specified, this value must be equal or less than the prefix range's prefix length plus 24, otherwise the server will use the prefix's range plus 24. Thus, a /32 prefix will only perform pool balancing for prefixes up to a /56.  
If unspecified, the lessor of the prefix range's prefix length plus 24 or 64 is used.  
Available prefix-delegation prefixes longer than this value will not be considered when balancing prefix-delegation pools.  
This value should specify be the same or smaller than the longest allowed prefix length to assure all prefix-delegation prefixes can be assigned to clients as balancing prefixes longer than this length would result in prefixes that can never be delegated to clients.

**name** [string](#) required,unique

Assigns a name to this prefix template.

**options-expr** [expr](#)

Defines an expression that evaluates to the list of embedded policy options to be created.

**owner** [nameref](#)(0)

Identifies the owner of this prefix, referenced by name. Owners can be used to limit administrative access to prefixes by owner.  
If the prefix has an associated link, the owner of the associated link will apply, if it is set. If there is no associated link, or the link owner is unset, the owner of the parent prefix will apply, if the prefix owner is unset.

**policy** [nameref](#)(0)

Refers to a shared policy to use when replying to clients.

**prefix-description-expr** [expr](#)

Defines an AT STRING expression to apply to the description on the prefix object created when using the template.

**prefix-name-expr** [expr](#)

Defines an expression that evaluates to an AT STRING value to use for the name of the prefix object created when using the template.

**range-end-expr** [expr](#)

Defines an expression that evaluates to the range-end for the prefix.

**range-expr** [expr](#)

Defines an expression that evaluates to an AT\_PREFIX value for the prefix range to be created.

**range-start-expr** [expr](#)

Defines an expression that evaluates to the range-start for the prefix.

**region** [nameref](#)(0)

Identifies the region for this prefix, referenced by name. Regions can be used to limit administrative access to prefixes by region.  
If the prefix has an associated link, the region for the associated link will apply, if it is set. If there is no associated link, or the link region is unset, the region for the parent prefix will apply, if the prefix region is unset.

**restrict-to-admin-allocation** [bool](#) default = disabled

Controls whether the prefix is restricted to administrative requests to allocate the next available address. If set, the server will only respond to a client with an address from this prefix if it has been pre-allocated to the client.

**restrict-to-reservations** [bool](#) default = disabled

Controls whether the prefix is restricted to client (or lease) reservations. If enabled, the DHCP server will not automatically assign addresses or delegate prefixes to clients but instead requires the address or prefix to be supplied by a

reservation, either a lease reservation or a client reservation, which is specified via a client entry or through an extension and the environment dictionary.

**reverse-zone-prefix-length** [rangeint](#)(0-124)

Identifies the prefix length of the reverse zone for ip6.arpa updates. The server forms the zone name using this value if configured; otherwise the prefix length is determined from the Prefix. This value must be a multiple of 4 as ip6.arpa zones are on 4 bit (nibble) boundaries. If not a multiple of 4, it is rounded up to the next higher multiple of 4.

**selection-tags** [nlist](#)(obj(0))

Associates selection tags with an IPv6 prefix.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## prefix-template-policy

`prefix-template-policy` - Edits the DHCP policy embedded in a prefix-template

### Synopsis

```
prefix-template-policy <name> delete
prefix-template-policy <name> set <attribute>=<value>
    [<attribute>=<value> ...]
prefix-template-policy <name> get <attribute>
prefix-template-policy <name> disable <attribute>
prefix-template-policy <name> enable <attribute>
prefix-template-policy <name> show

prefix-template-policy <name> setV6Option <opt-name | id>[.<instance>]
    <value> [-blob | -expression] [-roundrobin]
prefix-template-policy <name> addV6Option <opt-name | id>[.<instance>]
    <value> [-blob | -expression] [-roundrobin]
prefix-template-policy <name> getV6Option <opt-name | id>[.<instance>]
prefix-template-policy <name> unsetV6Option <opt-name | id>[.<instance>]
prefix-template-policy <name> listV6Options

prefix-template-policy <name> setV6VendorOption <opt-name | id>
    <opt-set-name> <value> [-blob | -expression]
prefix-template-policy <name> getV6VendorOption <opt-name | id>
    <opt-set-name> <opt-set-name>
prefix-template-policy <name> unsetV6VendorOption <opt-name | id>
    <opt-set-name> <opt-set-name>
prefix-template-policy <name> listV6VendorOptions
```

### Description

The `prefix-template-policy` command lets you configure a DHCP policy embedded in a DHCP prefix template. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case a prefix template. A `prefix-template-policy` is created implicitly when you first reference it, and is deleted when the `prefix-template` is deleted.

See the help file for the `policy` command for more information.

### Examples

```
nrcmd> prefix-template-policy exampleprefix set foward-zone-name=example.com.
nrcmd> prefix-template-policy exampleprefix enable allow-rapid-commit
```

### Status

### See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [scope-policy](#), [scope-template-policy](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the inhibit-all-renews attribute or the inhibit-renews-at-reboot attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

### allow-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses. The default is to allow clients to request temporary addresses.

### default-prefix-length [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it. The default length must always be greater than or equal to the prefix length of the prefix range.

### excluded-prefix [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603). Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified. Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64. If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

### forward-dnsupdate [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

### forward-zone-name [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.



Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action). This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet. Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases.



The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted.  
Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

- |   |          |   |
|---|----------|---|
| 1 | allow    | Allows clients to request reconfiguration support and the server will honor the request (default).  |
| 2 | disallow | Allows clients to request reconfiguration support but the server will not honor the clients' request.   |
| 3 | require  | Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option. |

This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

**reconfigure-via-relay** [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages.

If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

**reverse-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

**server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent

communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### **shortest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### **split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### **tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### **unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system\_default\_policy, then the default is 86400 seconds (or 24 hours).

#### **use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### **v4-bootp-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all BOOTP clients.

#### **v4-reply-options** [nlist](#)(obj(0))

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### **v6-max-client-preferred-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### **v6-max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### **v6-max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### **v6-max-client-valid-lifetime** [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### **v6-reply-options** [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy

processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

**v6-rsoo-allowed-options** [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

**valid-lifetime** [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime.  
Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, It overrides the valid-lifetime.

**view-id** [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## policy

policy - Specifies DHCP policy information

### Synopsis

```
policy <name> create [<attribute>=<value>]
policy <name> create clone=<clone-name>
policy <name> delete
policy list
policy listnames
policy listbrief
policy <name> set <attribute>=<value> [<attribute>=<value> ...]
policy <name> get <attribute>
policy <name> unset <attribute>

policy <name> disable <attribute>
policy <name> enable <attribute>
policy <name> show

policy <name> setLeaseTime <time-val>
policy <name> getLeaseTime

policy <name> setOption <opt-name | id> <value>
                        [-blob | -expression] [-roundrobin]
policy <name> getOption <opt-name | id>
policy <name> unsetOption <opt-name | id>
policy <name> listOptions

policy <name> setV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
policy <name> addV6Option <opt-name | id>[.<instance>]
                        <value> [-blob | -expression] [-roundrobin]
policy <name> getV6Option <opt-name | id>[.<instance>]
policy <name> unsetV6Option <opt-name | id>[.<instance>]
policy <name> listV6Options

policy <name> setVendorOption <opt-name | id> <opt-set-name>
                        <value> [-blob | -expression]
policy <name> getVendorOption <opt-name | id> <opt-set-name>
policy <name> unsetVendorOption <opt-name | id> <opt-set-name>
policy <name> listVendorOptions

policy <name> setV6VendorOption <opt-name | id> <opt-set-name>
                        <value> [-blob | -expression]
policy <name> getV6VendorOption <opt-name | id> <opt-set-name>
policy <name> unsetV6VendorOption <opt-name | id> <opt-set-name>
policy <name> listV6VendorOptions

policy < <name> | all > pull < ensure | replace | exact >
                        <cluster-name> [-report-only | -report]
policy < <name> | all > push < ensure | replace | exact >
```

```
                                <cluster-list> [-report-only | -report]
policy <name> reclaim <cluster-list> [-report-only | -report]
```

## Description

The policy command manages DHCP policy configurations. A policy is a collection of DHCP option values to associate with a range of addresses in a scope, or with a specific client or client-class configuration. Network Registrar considers policy reply options in a hierarchy of options. For details on these reply options, see the User Guide for Network Registrar.

The policy command by itself is for a named policy. You can also manage the embedded policies for dhcp-address-block, client, client-class and scope objects through the dhcp-address-block-policy, client-policy, client-class-policy, and scope-policy commands, respectively.

For embedded policies, name identifies the object that contains the embedded policy. For example, an attribute-setting command for a scope policy would be "scope-policy scope-name set <attribute>"--using the name of the scope for the name value.

The default policy is a special named policy that includes default settings. You can manage the default policy just like all the other named ones.

```
policy <name> setLeaseTime <time-val>
policy <name> getLeaseTime
```

Use the setLeaseTime command to set the values of lease times and the getLeaseTime command to display the value of a lease time.

```
policy <name> setV6Option <opt-name | id>[.<instance>]
                                <opt-set-name> <value> [-blob | -expression] [-roundrobin]
policy <name> addV6Option <opt-name | id>[.<instance>]
                                <opt-set-name> <value> [-blob | -expression] [-roundrobin]
policy <name> getV6Option <opt-name | id>[.<instance>]
                                <opt-set-name>
policy <name> unsetV6Option <opt-name | id>[.<instance>]
policy <name> listV6Options
```

To manage the DHCPv6 options on the policy, use the commands: setV6Option, addV6Option, getV6Option, unsetV6Option, listV6Options.

As some DHCPv6 options allow multiple instances to be configured, the <instance> value (1..n) can be specified on some commands to set (update), add as, get, or remove a specific instance of the option. Note that for unset, if multiple instances exist, they are all removed unless a specific index is specified. The instance can only be specified for those options that are defined to allow multiple instances to exist.

```
policy <name> setVendorOption <opt-name | id> <opt-set-name> <value>
                                [-blob | -expression]
policy <name> getVendorOption <opt-name | id> <opt-set-name>
policy <name> unsetVendorOption <opt-name | id> <opt-set-name>
policy <name> listVendorOptions
```

The setVendorOption and getVendorOption commands are used to set and get vendor-specific option data on the policy. These commands require an option name and the name of a vendor-specific option definition set.

The unsetVendorOption command removes the data for the specific vendor option.

The listVendorOptions command displays all vendor-option data that is set in the policy. The listing includes the name of the option-definition set that was used to define the data.

You can manage the DHCPv6 vendor options on the policy using the commands: setV6VendorOption, getV6VendorOption, unsetV6VendorOption, listV6VendorOptions.

### Note:

The setOption, setV6Option, addV6Option, setVendorOption, and setV6VendorOption commands support an optional "-blob" or "-expression" argument (after the option value). If "-blob" is specified, the option value is assumed to be in blob (hexadecimal) format regardless of how the option is defined. If "-expression" is specified, the option value is assumed to be a DHCP server expression which the server will evaluate for each request. As expressions are not easy to enter directly in the CLI, use @<file>

to read the expression from the named file.

The `setOption`, `setV6Option`, and `addV6Option` commands support an additional `"-roundrobin"` argument (after the option value) which, when specified, instructs the DHCP server to return option data which contains more than one value in a different, rotated, order. A particular client will always get the same order, but different clients will get different "rotations" of the order of the multiple values configured for the option based on their client identifier.

The `pull`, `push`, and `reclaim` commands are only available when connected to a regional cluster. For `push` and `reclaim`, a list of clusters or `"all"` may be specified.

## Examples

```
nrcmd> policy default setOption dhcp-lease-time 608400
nrcmd> policy default listOptions
      (51)dhcp-lease-time: 604800
nrcmd> policy default set grace-period=3d
```

## Status

## See Also

[option-set](#), [option](#)

## Attributes

### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the `inhibit-all-renews` attribute or the `inhibit-renews-at-reboot` attribute.

### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the `FQDN` option to indicate that it wants to do the A record update in the request, and if this value is `TRUE`, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to `false` (the default). Even if set to `true`, clients can request only lease times that are shorter than those configured for the server.

### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a `Discover/Solicit` with the `Rapid Commit` option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to `FALSE`, Rapid Commit is not allowed.
- If at least one has it set to `TRUE`, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are

checked.  
The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses.  
The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it.  
The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603).  
Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified.  
Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64.  
If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebind operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used.  
The default is the value of the default-prefix-length.  
This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this

value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

#### **max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

#### **max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

#### **max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

#### **offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

#### **packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

#### **packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

#### **packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

#### **permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

#### **preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, It overrides the preferred-lifetime.

#### **reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

1 allow	Allows clients to request reconfiguration support and the server will honor the request (default).
2 disallow	Allows clients to request reconfiguration support but the server will not honor the clients' request.
3 require	Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not



include a Reconfigure-Accept option. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

#### reconfigure-via-relay [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

#### reverse-dnsupdate [nameref\(0\)](#)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### server-lease-time [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### shortest-prefix-length [rangeint\(0-128\)](#)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### split-lease-times [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### unavailable-timeout [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system default\_policy, then the default is 86400 seconds (or 24 hours).

#### use-client-id-for-reservations [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### v4-bootp-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

#### v4-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### v6-max-client-preferred-lifetime [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value.



This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### v6-max-client-rebinding-time [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### v6-max-client-renewal-time [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### v6-max-client-valid-lifetime [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsoo-allowed-options [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## region

region - Configures a named geographic region

### Synopsis

```
region <tag> create <name> [<attribute>=<value>]
region <tag> delete
region list
region listnames
region listbrief
region <tag> show
region <tag> set <attribute>=<value> [<attribute>=<value> ...]
region <tag> get <attribute>
region <tag> unset <attribute>

region <tag> enable <attribute>
region <tag> disable <attribute>
```

```
region < <tag> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
region < <tag> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
region <tag> reclaim <cluster-list> [-report-only | -report]
```

## Description

The region command configures objects representing a geographic region containing other objects, such as address blocks, subnets and zones.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

## Status

## See Also

### Attributes

**contact** [string](#)

The contact information for this region.

**name** [string](#)

The full name or printable name for this region.

**tag** [string](#) required,unique

The unique tag name for this region. Typically a short name referring to this region.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## report

report - Creates a summary report of address usage

## Synopsis

```
report [column-separator=<string>]
    [dhcp-only]
    [dhcpv4 [-network]]
    [dhcpv6 [-network]]
    [file=<output file>]
    [vpn=<vpn-name>]
```

## Description

The report command provides address utilization reporting for all subnets containing addresses that are handled by the DHCP server. The DHCP server must be running to obtain the report or a "Connection Failure" error message will be returned.

The report command provides v4 and v6 utilization. The option 'dhcp-only' is provided as a synonym for 'dhcpv4', to support backwards compatibility. The '-network' option can be used to sort the report in network order. With this option, the v4 report is sorted by primary-subnet, and subtotals by selection tag are included. Subtotals for parent address blocks are omitted.

The v6 report is sorted by link, with subtotals by selection tag for dhcp and prefix-delegation prefixes. Subtotals for parent prefixes are omitted.

The output for the report dhcpv4 command is a table consisting of column-aligned data for each defined scope, subnet, and CCM address block. Subtotals are listed for subnets and address blocks that contain multiple scopes or subnets. A grand total row summarizes the data from all the subnets.

Each row lists the following information:

- o Subnet/Mask
- o Scope Name
- o %Free - percentage of dynamic addresses available to be leased
- o Total Dynamic - total number of addresses configured, excluding reservations
- o Total Reserved - total reserved addresses
- o Leased - addresses actively leased by clients
- o Avail - addresses available to be leased
- o Other Avail - addresses set aside to be leased by this server's failover partner
- o Pending Avail - leases not available to be leased because the server is in the communications-interrupted failover state.
- o In Transition - leases offered to clients or waiting for the configured grace period before again becoming available
- o Reserved Active - reserved addresses actively leased by clients
- o Unavail - addresses marked unavailable
- o Active Deactivated - leased addresses that have been administratively deactivated
- o Deactivated - addresses that have been administratively deactivated

The output for the report dhcpv6 command is a table consisting of column-aligned data for each defined prefix. If requested, totals will be computed for each parent prefix defined in the address tree. Separate total lines will be shown for child prefixes classified for prefix delegation or dhcp allocation. No counters apply for stateless configuration or infrastructure entries. The only attribute that may be reported is whether the prefix has been deactivated. Total lines will be skipped if the prefix-count for a given type is 0.

Each row lists the following information:

- o Prefix/Mask
- o Prefix Name
- o Total Reserved - total number of reserved leases configured in this prefix
- o Leased - addresses actively leased by clients
- o Revoked - dynamic leases (leases that are not reserved) that are no longer usable by the client, but that may still be in use
- o In Transition - leases offered to clients or waiting for the configured grace period before again becoming available
- o Reserved Active - reserved addresses actively leased by clients
- o Reserved Inactive - reserved addresses not used by their assigned DHCP clients
- o Reserved Unavail - reserved addresses marked unavailable
- o Unavail - addresses marked unavailable
- o Active Dynamic - total number of dynamic leases actively in use
- o Active Deactivated - number of dynamic and reserved leases currently leased by DHCP clients, but have also been administratively deactivated
- o Deactivated - addresses that have been administratively deactivated

## Examples

```
report
report file=myreport.txt
```

## Status

## See Also

[export](#) addresses, [lease-notification](#), [session](#) current-vpn

## Report Keywords

**column-separator**  
Specifies the character string you want used between the columns in the report. The default is a single space. If you specify more than one space, you must use a backslash (\) to allow the extra spaces, and if you enter the spaces on the command line, use quotation marks.

**dhcp-only**  
Provided for command-syntax compatibility with prior versions, specifies a summary of the DHCP server information. This is the only option available for the report command, and is no longer required to run the command.

**dhcpv4**  
Displays ipv4 utilization.

**dhcpv6**  
Display ipv6 utilization.

**file**  
Specifies the filename to which the report command writes the output. If you do not specify a filename, the report command writes to 'standard out'.

**vpn**  
The VPN address space from which to select scopes to examine when executing this command. If no vpn-name is specified, then the session's current-vpn is used. If the reserved vpn-name "global" is used, then the global (or unnamed) VPN address space is used. The reserved vpn-name "all" is not allowed for this command, because the report command has no mechanism to distinguish identical IP addresses in different VPNs.

## reservation

reservation - Configures DHCPv4 reservations

### Synopsis

```
reservation [<vpn-name>/]<ipaddr> create (<macaddr>|<lookup-key>)
[-mac|-blob|-string] [<attribute>=<value>...]
reservation [<vpn-name>/]<ipaddr> delete
reservation [<vpn-name>/]<ipaddr> get <attribute>
reservation [<vpn-name>/]<ipaddr> set <attribute>=<value>...
reservation [<vpn-name>/]<ipaddr> unset <attribute>
reservation [<vpn-name>/]<ipaddr> show
reservation list [[<vpn-name>/]<ipaddr>|-mac|-key]
reservation listbrief [-macaddr=<mac-addr>]
[-lookup-key=<lookup-key>] [-mac|-blob|-string]
[-vpn=<vpn-name>] [-count-only]
```

### Description

The reservation command lets you manipulate Network Registrar's global list of reservations. Changes to reservations also modify the reservations listed with each scope (and vice versa). Thus, these commands are an alternative to the scope's addReservation, removeReservation, and listReservation commands.

A matching scope must exist for each reservation in the global list. When you create or delete reservations, they are added or removed from the appropriate scope. When multiple scopes exist for the same subnet, you must specify the include-tags attribute to match the reservation to the appropriate scope or specify the scope (scope=<name>). If no matching scope is found, the edit is rejected as invalid.

The reservation list command displays the reservations in address order unless -mac or -key is specified to change the sort order.

## Examples

## Status

## See Also

### Attributes

#### client-class [nameref\(0\)](#)

Identifies the client-class to use when selecting from among scopes that might contain a specified reservation. Used when creating a reservation. This attribute is ignored if the reservation is already assigned to a scope.

#### cm-mac-address [macaddr](#)

Sets the contents of DHCP option-82 (relay-agent-info) remote-id (sub-option 2) to be the value of the cable-modem's MAC address. When using the CMTS source-verify capability, a DHCPv4 leasequery response must contain a valid option-82 with the cable-modem's MAC address in the remote-id (sub-option 2). If you configure the cable-modem's MAC address in this attribute, it will be used to create the remote-id in a relay-agent-info option whenever there isn't a relay-agent-info option available from the lease state database reflecting an actual relay-agent-info option sent in by an actual DHCP client. This would be the case if the lease was statically allocated.

#### description [string](#)

Describes the device that this reservation object represents.

#### device-name [string](#)

Displays the name of the device represented by this reservation object.

#### include-tags [nlist\(obj\(0\)\)](#)

Displays the selection criteria for this reservation. Used when creating a reservation to select the scope (when no scope has been specified) or to validate that the correct scope has been specified. This attribute is ignored if the reservation is already assigned to a scope.

#### ipaddr [ipaddr](#) required,immutable

Displays an IP address within the network that the scope specifies and that contains the reservation.

#### lookup-key [blob](#)

Specifies the sequence of bytes that is the key for this reservation object. The type for this key is set in the lookup-key-type attribute. The string representation of this key is defined by its associated lookup-key-type parse and unparse methods. For example, a mac address key would be converted from a string to raw form with the AT\_MACADDR parse() method, and converted from raw form to a string by the AT\_MACADDR unparse() method.

#### lookup-key-type [int](#)

Specifies the data type for the value in the lookup-key attribute. The type may take on the value of 7 for AT\_BLOB, 9 for AT\_MACADDR, or 46 for AT\_NSTRING.

#### scope [nameref\(0\)](#)

Identifies the scope for this reservation. On local clusters, a parent scope must exist and this attribute must always be set. On regional clusters, this attribute may be unset, pending a push operation to the local cluster.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### vpn-id [int](#) default = 0, immutable

Displays the identifier of the VPN that contains this reservation.

reservation6 - Configures DHCPv6 reservations

## Synopsis

```
reservation6 [<vpn-name>/]<address> create <lookup-key>
[-blob|-string] [<attribute>=<value>...]
reservation6 [<vpn-name>/]<address> delete
reservation6 [<vpn-name>/]<address> get <attribute>
reservation6 [<vpn-name>/]<address> set <attribute>=<value>...
reservation6 [<vpn-name>/]<address> unset <attribute>
reservation6 [<vpn-name>/]<address> show
reservation6 list [<vpn-name>/]<address>|-key]
reservation6 listbrief [-lookup-key=<lookup-key> [-blob|-string]]
[-vpn=<vpn-name>] [-count-only]
```

## Description

The reservation6 command lets you manipulate Network Registrar's global list of DHCPv6 reservations. Changes to reservations also modify the reservations listed with each prefix (and vice versa). Thus, these commands are an alternative to the prefix's addReservation, removeReservation, and listReservation commands.

A matching prefix must exist for each reservation in the global list. When you create or delete reservations, they are added or removed from the appropriate prefix. When multiple prefixes exist for the same prefix, you must specify the include-tags attribute to match the reservation to the appropriate prefix or specify the prefix name (prefix=<name>). If no matching prefix is found, the edit is rejected as invalid.

The reservation6 list command displays the reservations in address order unless -key is specified to change the sort order.

## Examples

## Status

## See Also

### Attributes

client-class [nameref\(0\)](#)

Identifies the client class to use when selecting from among scopes that might contain a specified reservation. Used when creating a reservation. This attribute is not persistent in the database.

cm-mac-address [macaddr](#)

Specifies the cable-modem MAC address value to be used in generating the CableLabs vendor option (17) with the cm-mac-address option (1026) as the leasequery relay data (lq-relay-data) if no relay data is otherwise available for the lease (such as for clients that do not perform DHCP). This information is critical for clients that use a statically configured address when using the CMTS source-verify feature.

description [string](#)

Describes the device that this reservation object represents.

device-name [string](#)

Displays the name of the device represented by this reservation object.

include-tags [nlist\(obj\(0\)\)](#)

Displays the selection criteria for this reservation. Used when creating a reservation to select the scope (when no scope has been specified) or to validate that the correct scope has been specified. This attribute is not persistent in the database.

**ip6address** [ip6](#) required,immutable

Specifies the IPv6 address for the reservation.

**lookup-key** [blob](#) required

Specifies the sequence of bytes that is the key for this reservation object. The type for this key is set in the lookup-key-type attribute. The string representation of this key is defined by its associated lookup-key-type parse and unparse methods. For example, a blob key would be converted from a string of colon-separated hex digits to raw form with the `AT_BLOB parse()` method, and converted from raw form to a string by the `AT_BLOB unparse()` method.

**lookup-key-type** [int](#) required

Specifies the data type for the value in the lookup-key attribute. The type may take on the value of 7 for `AT_BLOB` or 46 for `AT_NSTRING`.

**prefix** [nameref\(0\)](#)

Identifies the prefix for this reservation.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**vpn-id** [int](#) default = 0, immutable

Displays the identifier of the VPN that contains this reservation.

## resource

`resource` - configures resources limits and allows for viewing and resetting resources

### Synopsis

```
resource set <attribute>=<value> [<attribute>=<value> ...]
resource get <attribute>
resource unset <attribute>
resource enable <attribute>
resource disable <attribute>
resource [show]

resource report [all | full | levels]

resource reset [name[,name[,...]]]
```

### Description

The `resource` command is used to configure resource warning and critical limits, display the current status of monitored resources and reset the peak values.

The `resource report` command lists the resources that are above their (configured or default) critical or warning levels. If all is specified, all resources are listed regardless of their state. If full is specified, the complete objects are reported. If levels is specified, the current warning and critical levels in effect for each resource are reported.

The `resource reset` command resets all resources or just the optionally named resources.

### Examples

### Status

### See Also

### Attributes

#### **ccm-memory-critical-level** [filesz](#)

Specifies the critical level for the CCM server's memory usage in bytes. If the CCM server's virtual memory size exceeds this value, a critical notification is triggered.  
If not specified, the default-memory-critical-level attribute value is used.

#### **ccm-memory-warning-level** [filesz](#)

Specifies the warning level for the CCM server's memory usage in bytes. If the CCM server's virtual memory size exceeds this value, a warning notification is triggered.  
If not specified, the default-memory-warning-level attribute value is used.

#### **cdns-memory-critical-level** [filesz](#)

Specifies the critical level for the Caching DNS server's memory usage in bytes. If the DNS server's virtual memory size exceeds this value, a critical notification is triggered.  
If not specified, the default-memory-critical-level attribute value is used.

#### **cdns-memory-warning-level** [filesz](#)

Specifies the warning level for the Caching DNS server's memory usage in bytes. If the DNS server's virtual memory size exceeds this value, a warning notification is triggered.  
If not specified, the default-memory-warning-level attribute value is used.

#### **cdns-security-events-critical-level** [int](#) default = 75

Specifies the critical level for the number of DNS security events in the Caching DNS server. If the server's value for security-events-periodic exceeds this value, a critical notification is triggered.

#### **cdns-security-events-warning-level** [int](#) default = 50

Specifies the warning level for the number of DNS security events in the Caching DNS server. If the server's value for security-events-periodic exceeds this value, a warning notification is triggered.

#### **certificate-expiration-critical-level** [percent](#) default = 10%

Specifies the critical level for the Certificate's expiration. If the current time exceeds this value, a critical notification is triggered.

#### **certificate-expiration-warning-level** [percent](#) default = 25%

Specifies the warning level for the Certificate's expiration. If the current time exceeds this value, a warning notification is triggered.

#### **cnrservagt-memory-critical-level** [filesz](#)

Specifies the critical level for the cnrservagt's memory usage in bytes. If the cnrservagt's virtual memory size exceeds this value, a critical notification is triggered.  
If not specified, the default-memory-critical-level attribute value is used.

#### **cnrservagt-memory-warning-level** [filesz](#)

Specifies the warning level for the cnrservagt's memory usage in bytes. If the cnrservagt's virtual memory size exceeds this value, a warning notification is triggered.  
If not specified, the default-memory-warning-level attribute value is used.

#### **data-free-space-critical-level** [rangeint](#)(1-10) default = 2

Specifies the critical level multiplier for the data volume's free space. If the data volume free space falls below this number of times the last shadow backup size, a critical notification is triggered.

#### **data-free-space-warning-level** [rangeint](#)(1-10) default = 4

Specifies the warning level multiplier for the data volume's free space. If the data volume free space falls below this number of times the last shadow backup size, a warning notification is triggered.

#### **default-memory-critical-level** [filesz](#) transient

Provides the default critical level for memory used if none is specified for a particular server. This value is dependent on the operating system and is 87.5% of the memory limited determined to be reasonable for Network Registrar processes.

#### **default-memory-warning-level** [filesz](#) transient

Provides the default warning level for memory used if none is specified for a particular server. This value is dependent on the operating system and is 75% of the memory limit determined to be reasonable for Network Registrar processes.



#### **dhcp-memory-critical-level** [filesz](#)

Specifies the critical level for the DHCP server's memory usage in bytes. If the DHCP server's virtual memory size exceeds this value, a critical notification is triggered. If not specified, the default-memory-critical-level attribute value is used.

#### **dhcp-memory-warning-level** [filesz](#)

Specifies the warning level for the DHCP server's memory usage in bytes. If the DHCP server's virtual memory size exceeds this value, a warning notification is triggered. If not specified, the default-memory-warning-level attribute value is used.

#### **dns-memory-critical-level** [filesz](#)

Specifies the critical level for the Authoritative DNS server's memory usage in bytes. If the DNS server's virtual memory size exceeds this value, a critical notification is triggered. If not specified, the default-memory-critical-level attribute value is used.

#### **dns-memory-warning-level** [filesz](#)

Specifies the warning level for the Authoritative DNS server's memory usage in bytes. If the DNS server's virtual memory size exceeds this value, a warning notification is triggered. If not specified, the default-memory-warning-level attribute value is used.

#### **dns-security-events-critical-level** [int](#) default = 75

Specifies the critical level for the number of DNS security events in the Authoritative DNS server. If the server's value for security-events-periodic exceeds this value, a critical notification is triggered.

#### **dns-security-events-warning-level** [int](#) default = 50

Specifies the warning level for the number of DNS security events in the Authoritative DNS server. If the server's value for security-events-periodic exceeds this value, a warning notification is triggered.

#### **lease-count-critical-level** [int](#) default = 8000000

Specifies the critical level for the number of DHCP leases. If the DHCP server's DHCPv4 configured plus DHCPv6 allocated leases exceeds this value, a critical notification is triggered.

#### **lease-count-warning-level** [int](#) default = 6000000

Specifies the warning level for the number of DHCP leases. If the DHCP server's DHCPv4 configured plus DHCPv6 allocated leases exceeds this value, a warning notification is triggered.

#### **queued-binding-updates-critical-level** [percent](#) default = 25%

Specifies the critical level for queued-binding-updates. If the current number of queued binding updates exceeds this percentage of the number of DHCP leases, a critical notification is triggered. Note: The minimum value is queued-binding-updates-warning-level.

#### **queued-binding-updates-warning-level** [percent](#) default = 10%

Specifies the warning level for queued-binding-updates. If the current number of queued binding updates exceeds this percentage of the number of DHCP leases, a warning notification is triggered. Note: The minimum value is 1000 binding updates.

#### **rr-count-critical-level** [int](#) default = 25000000

Specifies the critical level for the number of resource records (RRs) in the authoritative DNS server. If the DNS server's RR count exceeds this value, a critical notification is triggered.

#### **rr-count-warning-level** [int](#) default = 20000000

Specifies the warning level for the number of resource records (RRs) in the authoritative DNS server. If the DNS server's RR count exceeds this value, a warning notification is triggered.

#### **shadow-backup-time-critical-level** [time](#) default = 12h

Specifies the critical level for the nightly backup's elapsed time. If the nightly backup takes longer than this time, a critical notification is triggered.

#### **shadow-backup-time-warning-level** [time](#) default = 6h

Specifies the warning level for the nightly backup's elapsed time. If the nightly backup takes longer than this time, a warning notification is triggered.

#### **snmp-memory-critical-level** [filesz](#)

Specifies the critical level for the SNMP server's memory usage in bytes. If the SNMP server's virtual memory size exceeds this value, a critical notification is triggered. If not specified, the default-memory-critical-level attribute value is used.

#### **snmp-memory-warning-level** [filesz](#)

Specifies the warning level for the SNMP server's memory usage in bytes. If the SNMP server's virtual memory size exceeds this value, a warning notification is triggered. If not specified, the default-memory-warning-level attribute value is used.

#### **tftp-memory-critical-level** [filesz](#)

Specifies the critical level for the TFTP server's memory usage in bytes. If the TFTP server's virtual memory size exceeds this value, a critical notification is triggered. If not specified, the default-memory-critical-level attribute value is used.

#### **tftp-memory-warning-level** [filesz](#)

Specifies the warning level for the TFTP server's memory usage in bytes. If the TFTP server's virtual memory size exceeds this value, a warning notification is triggered. If not specified, the default-memory-warning-level attribute value is used.

#### **tomcat-memory-critical-level** [filesz](#)

Specifies the critical level for the Tomcat server's memory usage in bytes. If the Tomcat server's virtual memory size exceeds this value, a critical notification is triggered. If not specified, the default-memory-critical-level attribute value is used.

#### **tomcat-memory-warning-level** [filesz](#)

Specifies the warning level for the Tomcat server's memory usage in bytes. If the Tomcat server's virtual memory size exceeds this value, a warning notification is triggered. If not specified, the default-memory-warning-level attribute value is used.

#### **trap-interval** [time](#) default = 60s

Specifies the minimum interval between traps for each monitored resource.  
Note: The minimum interval possible depends on the CCMServerConfig stats-history-sample-interval attribute (which defaults to 10s), as this is the interval at which the resource limits are checked.

#### **trap-repeat-interval** [time](#) default = 15m

Specifies the minimum interval between traps for when a resource is in the critical or warning state, with a possibly updated current value for the resource.  
Note: If this value is less than trap-interval, trap interval is used.

#### **traps-enabled** [bool](#) default = enabled

Specifies whether to send SNMP traps for resource notifications. Traps will be sent when a value exceeds the critical or warning levels, or if the value previously exceeded the levels and has dropped below the warning level.  
Note that when a resource is above the critical or warning level, traps are sent out periodically until the resource drops below the warning level.

#### **zone-count-critical-level** [int](#) default = 10000

Specifies the critical level for the number of zones in the authoritative DNS server. If the DNS server's zone count exceeds this value, a critical notification is triggered.

#### **zone-count-warning-level** [int](#) default = 8000

Specifies the warning level for the number of zones in the authoritative DNS server. If the DNS server's zone count exceeds this value, a warning notification is triggered.

## role

role - Configures a role

### Synopsis

```
role <name> create <base-role> [<attribute>=<value>]
role <name> delete
role list
role listnames
role listbrief
role <name> show
```

```

role <name> set <attribute>=<value> [<attribute>=<value> ...]
role <name> get <attribute>

role <name> enable <attribute>
role <name> disable <attribute>

role <name> addConstraint [<attribute>=<value> ...]
role <name> Constraint <index> set [<attribute>=<value> ...]
role <name> Constraint <index> unset [<attribute> ...]
role <name> removeConstraint <index>
role <name> listConstraints

role < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
role < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-omitrelated] [-report-only | -report]
role <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The role command configures the specified role. A role describes the operations that an administrator can perform and any data constraints that should be applied. A role must be assigned to an administrator group to be associated with an administrator.

The addConstraint, Constraint, removeConstraint, and listConstraint commands are used to manipulate the constraints.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

The push operation will also push the related groups (using replace mode) and related owners and regions (using ensure mode). The pull operation will pull related owners and regions (using ensure mode). For either operation, specify -omitrelated to prevent this and just push or pull the role.

## Examples

## Status

## See Also

[group](#), [admin](#)

## Attributes

**all-sub-roles** [bool](#) default = true

Controls whether to ignore the sub-role attribute for this attribute. If this attribute is unset, or if it is set to true, then the server ignores the value of the sub-roles attribute and this subrole is authorized for all sub-roles. If this attribute is false, then the sub-roles attribute provides the list of subroles for which this role instance is authorized. If the unconstrained attribute is set to true, then the values of this attribute and the sub-roles attribute are ignored, and the sub-role authorization for the role is applicable for all sub-roles.

**groups** [nlist\(obj\(0\)\)](#)

Lists the groups with which this role is associated. Any member of a listed group can perform the operations that the role allows.

**name** [string](#) required,unique

Identifies the name of this role.

**read-only** [bool](#) default = false

Indicates that all constraints associated with this role are limited to read-only access.

**role** [string](#)

Specifies the base role for this object. The base role defines operations, such as modifying a zone, that are allowed and the further constraints on these operations. For example,

a constrained role could limit the list of zones to a specific list of Owners.

#### sub-roles [nlist](#)(obj(0))

Lists subroles associated with this role instance. If the all-sub-roles attribute is unset, or if it is set to true, then this attribute is ignored. If the all-sub-roles attribute is set to false, then this attribute specifies the list of subroles for this role instance, and an administrator associated with this role has authorization limited to the specified subroles. If the administrator has multiple roles in which the role attribute is the same, then subrole authorization for that role should be taken to be the union of all the sets of subroles from the individual role instances; and, if any of these role instances has the all-sub-roles attribute set to true, then subrole authorization for that role is for all sub-roles. Also, if any role instance for a matching role has the unconstrained attribute set to true, then subrole authorization for that role is for all subroles.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### unconstrained [bool](#) default = false

Indicates that this role has no other constraints beyond the list of operations it can perform.

## router

router - Configures a router

### Synopsis

```
router <name> create <address> [<attribute>=<value> ...]
router <name> delete
router list
router listnames
router listbrief
router <name> show
router <name> get <attribute>
router <name> set <attribute>=<value> [<attribute>=<value> ...]
router <name> unset <attribute>

router <name> listInterfaces
```

### Description

The router command lets you configure a router.

Note: The router (and router-interface) objects can be used to record information about the network configuration; they are optional and informational only.

### Examples

### Status

### See Also

[router-interface](#)

### Attributes

address [ipaddr](#) unique

Sets the management interface address for this router.

description [string](#)

Describes this router.

interfaces [nlist](#)(obj(0)) transient

Lists objects describing the interfaces associated with this

router. This is used when the CCM server is returning a list of routers and interfaces for the UI to display a tree of routers and interfaces.

**ip6address** [ip6addr](#) unique

Sets the IPv6 management interface address for this router.

**name** [string](#) required,unique

Names this router.

**owner** [oid](#)

Controls the owner of this object. This owner field is used to group similarly owned routers and can be used to limit administrative access.

**region** [oid](#)

Controls the region associated with this object. This region field is used to group similarly located routers and can be used to limit administrative access.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## router-interface

router-interface - Configures an interface on a router

### Synopsis

```
router-interface <router> <name> create [<attribute>=...]  
  
router-interface <router> <name> delete  
router-interface <router> list  
router-interface <router> listnames  
router-interface <router> listbrief  
router-interface <router> <name> show  
  
router-interface <router> <name> get <attribute>  
router-interface <router> <name> set <attribute>=<value>  
router-interface <router> <name> set <attribute>=<value>  
[<attribute>=<value> ...]  
router-interface <router> <name> unset <attribute>
```

### Description

The router-interface command lets you configure an interface on a router.

### Examples

### Status

### See Also

[router](#)

### Attributes

**bundle-id** [int](#)

Specifies the bundle id for a group of bundled interfaces.

**cable-helper** [nlist](#)(obj(0))

Specifies the list of cable-helper IPv4 addresses on this interface.

**delegatable-prefixes** [nlist](#)(obj(0)) transient

Lists the prefix delegation IPv6 prefixes configured for this



```
router-login-template <name> unset <attribute>
router-login-template <name> get <attribute>
router-login-template <name> [show]
router-login-template list
router-login-template listnames
router-login-template listbrief
```

## Description

The `router-login-template` command lets you configure template to use when logging into router.

Note: This command is now obsolete, but is still available when connected to clusters running earlier releases.

## Examples

```
nrcmd> router-login-template example-template create login-prompt=> enable-prompt=#
nrcmd> router-login-template example-template delete
```

## Status

## See Also

[router](#), [router-interface](#)

## Attributes

`enable-password-prompt` [string](#)

The string that is used as the enable password prompt by the router.

`enable-prompt` [string](#) required

The string that is used as the prompt by the router in enable mode.

`login-prompt` [string](#) required

The string that is used as the login prompt by the router.

`name` [string](#) required,unique

The name of this template

`password-prompt` [string](#)

The string that is used as the user password prompt by the router.

`tenant-id` [short](#) default = 0, immutable

Identifies the tenant owner of this object.

`username-prompt` [string](#)

The string that is used as the username prompt by the router.

## router-type

`router-type` - Displays the available router types

## Synopsis

```
router-type list
router-type listnames
```

## Description

The `router-type` command displays the available router types.

Note: This command is now obsolete, but is still available when connected to clusters running earlier releases.

## Examples

## Status

## See Also

[router](#)

## Attributes

### description [string](#)

The description of this router type.

### java-class-name [string](#) required

The java class name for an implementation of `com.cisco.cnr.ricsrv.IRouter` that can communicate with this type of router.

### manufacturer [string](#)

The name of the manufacturer of this type of router.

### name [string](#) required,unique

The name of this router type.

### router-os-version [string](#)

The OS version, as a string for this type of router.

## save

```
save - Saves the current changes to the cluster
```

## Synopsis

```
save
```

## Description

The save command saves the current configuration changes to the database.

## Examples

## Status

## See Also

## scope

```
scope - Specifies the scope's properties
```

## Synopsis

```
scope list  
scope listnames  
scope listbrief
```



```

scope <name> create <address> <mask> [template=<template-name>]
                                [<attribute>=<value>...]
scope <name> delete
scope <name> set <attribute>=<value> [<attribute>=<value> ...]
scope <name> get <attribute>
scope <name> unset <attribute>
scope <name> disable <attribute>
scope <name> enable <attribute>
scope <name> show

scope <name> listLeases

scope <name> addRange <start> <end>
scope <name> removeRange <start> <end>
scope <name> listRanges

scope <name> addReservation <ipaddr> (<macaddr>|<lookup-key>)
                                [-mac | -blob | -string]
scope <name> removeReservation (<ipaddr>|<macaddr>|<lookup-key>)
                                [-mac | -blob | -string]
scope <name> listReservations

scope <name> clearUnavailable

scope <name> getUtilization

scope <name> applyTemplate <template-name>

scope report-staged-edits

```

## Description

The scope command lets you manipulate address ranges in the DHCP server.

When creating a scope using a template, specify - for the <name> to allow the scope template's scope-name to name the scope.

```

scope <name> listLeases
  The listLeases command lists the leases associated
  with this scope.

```

```

scope <name> addRange <start> <end>
scope <name> removeRange <start> <end>
scope <name> listRanges
  The addRange command adds a range of available addresses
  to the scope. The <start> and <end> values can either be
  full IP addresses or host numbers within the scope's
  subnet. It is an error if either <start> or <end> is
  an IP address outside the scope's subnet or is a host
  number that exceeds the bits allocated to host numbers
  by the scope's netmask.

```

The removeRange command removes a range of addresses from the scope's control.

The listRanges command lists the current ranges of addresses available for allocation.

```

scope <name> addReservation <ipaddr> (<macaddr>|<lookup-key>)
                                [-mac | -blob | -string]
scope <name> removeReservation (<ipaddr>|<macaddr>|<lookup-key>)
                                [-mac | -blob | -string]
scope <name> listReservations

```

The addReservation command adds a reservation for a specific IP address to a specific MAC address or lookup key.

The removeReservation command removes the reservation for an IP address, MAC address, or lookup key.

The listReservations command lists the available reservations.

```

scope <name> clearUnavailable
  The clearUnavailable command moves all unavailable leases
  in the specified scope to available.

```

```

scope <name> getUtilization
  The getUtilization command displays the current utilization
  object for the scope.

```

### scope <name> applyTemplate

The applyTemplate command applies the specified scope-template to the scope. All properties configured on the scope-template are applied to the scope.

### scope report-staged-edits

Displays a list of the scopes that have pending edits that have not been synchronized with the DHCP server.

Note: The scope command manages a VPN through a virtual attribute:

```
vpn [] (AT_STRING, Optional, default: <none>)
```

Use this attribute to set or get the VPN ID by VPN name instead of by ID. If not specified, the session's current vpn is used.

## Examples

```
nrcmd> scope scope-example create 192.168.0.0 255.255.255.224
nrcmd> scope scope-example addRange 192.168.0.33 192.168.0.62
nrcmd> scope scope-example addReservation 192.168.0.40 00:d0:ba:d3:bd:3b
```

## Status

## See Also

[scope-template](#)

## Attributes

**allocate-first-available** [bool](#) default = false

Enables you to force the allocation of new IP addresses from this scope to be the first available IP address; otherwise, the default of the 'least recently used' IP address is used. If this attribute is not set, or is unset, then the DHCP server attribute priority-address-allocation controls whether to allocate the first available IP address. If priority-address-allocation is set, and allocate-first-available for the scope is unset, then the scope allocates addresses as if allocate-first-available was set. If allocate-first-available is enabled or disabled for a scope, then for that scope the setting of priority-address-allocation has no meaning.

**allocation-priority** [int](#) default = 0

Assigns an order to scopes for allocating IP addresses. Acceptable scopes, with the highest allocation priority, grant IP addresses until the addresses are exhausted. You can mix scopes with an allocation-priority along with those without a priority in the same network. In this case, scopes with allocation priorities are examined for acceptability before those scopes with no allocation-priority. Lower numeric values have higher priorities, but an allocation-priority of 0 (the default) has no priority. If this attribute is not set, or is unset or 0, then the DHCP priority-address-allocation attribute controls the priority of the scope. If the DHCP priority-address-allocation attribute is set, and allocation-priority for the scope is unset, then the allocation-priority for the scope is the network number of the scope. If you explicitly set allocation-priority, then, for that scope, the DHCP setting of priority-address-allocation has no meaning.

**backup-pct** [percent](#)

Determines the percentage of available addresses that the main server sends to the backup server. If you define this value using the scope command, make sure you define it on the main server. If you define it on a backup server, it is ignored. Used with the scope command, the backup-pct attribute overrides the defined values on the failover pair for backup-pct and dynamic-bootp-backup-pct. The attribute value defined with the scope command becomes the value used for this scope, whether or not this scope supports dynamic-bootp. If you set the value to zero (0), the backup server receives no addresses. Since 0 is a significant value, once you set this value, you must unset it for the scope to use the failover pair's values for backup-pct or dynamic-bootp-backup-pct. Note: If the failover pair is configured to use load balancing, the backup-pct is ignored and 50% is used.

**bootp** [bool](#) default = disabled

Controls whether the server accepts BOOTP requests. If you want clients to always receive the same addresses, you must reserve IP addresses for all your BOOTP clients.

**deactivated** [bool](#)

Controls whether a scope extends leases to any clients. A deactivated scope does not extend leases to any clients. It treats all addresses in its ranges as if they were individually deactivated.

**description** [string](#)

Describes the scope.

**dhcp** [bool](#) default = enabled

Controls whether the DHCP server accepts DHCP requests for this scope. Disable DHCP if you want a scope to use BOOTP exclusively or you want to deactivate the scope temporarily.

**dns-host-bytes** [rangeint](#)(1-4)

Tells DHCP how many bytes in a lease IP address to use when forming in-addr.arpa names. The server forms names in the in-addr zone by prepending dns-host-bytes of IP address (in reverse order) to the reverse zone name. If unset, the server synthesizes an appropriate value based on the scope's subnet size.

**dynamic-bootp** [bool](#) default = disabled

Controls whether the server will accept dynamic BOOTP requests for this scope. Dynamic BOOTP requests are BOOTP requests that do not match a reservation, but could be satisfied from the available lease pool. To use this feature you must also enable bootp.

**embedded-policy** [obj](#)(0)

Displays the embedded policy for a scope.

**failover-backup-allocation-boundary** [ipaddr](#)

Sets the IP address allocation boundary for a backup server in a failover relationship. If the allocate-first-available attribute is set, the backup server allocates IP addresses in descending order from this boundary. If the allocate-first available attribute is unset or set to 0, the boundary used for allocating addresses is half the distance between the first and last IP address configured in the ranges for this scope. If no IP addresses are available below this boundary, the first IP address available above this boundary is used.

**free-address-config** [nameref](#)(0)

Identifies which trap captures unexpected free address events on this scope.

**ignore-declines** [bool](#) default = false

Determines whether the server reacts to server DHCPDECLINE messages that refer to one of the scope's IP addresses. If enabled (true), the DHCP server ignores all declines that refer to an IP address in this scope. If disabled, the DHCP server sets to UNAVAILABLE every IP address referred to in this scope. Default is false (disabled).

**maintenance** [bool](#) default = disabled

If this attribute is enabled, this Scope and any other Scopes on the same Link (Network) will participate in the current maintenance window. DHCP clients will have their renewal times adjusted to avoid renewals during the maintenance window. In addition, they will be directed to renew during the renewal-period after the end of the maintenance window.

**ping-clients** [bool](#)

If this attribute is not set, or is unset, then the DHCP server attribute ping-clients controls whether the server should attempt to ping an address before offering a lease. If enabled (true), this attribute also indicates a ping timeout. If ping-clients is set and ping-clients for the scope is unset, then the server pings an address as if it was set in the scope. If ping-clients is enabled or disabled for a scope, then for that scope the setting of ping-clients at the server level has no meaning. If not specified for the scope, the DHCP server's 'ping-clients' is used as the default.

**ping-timeout** [int](#)

Sets the number of milliseconds the DHCP server waits for ping responses. If you make this value too large, you slow down the lease offering processes. If you make this value too small, you reduce the effectiveness of pinging addresses before offering them. 300 milliseconds (the default value) is often the best choice. Only used if 'ping-clients' is enabled either for this scope or for the DHCP server. If not specified for the scope, the DHCP server's 'ping-timeout' is used as the default.

policy [nameref](#)(0) default = default

Identifies the name of the policy associated with this scope. Default is the default policy. This means that the scope uses all the properties set in the default policy (including the lease time), unless you specifically reset a property.

primary-subnet [subnet](#)

Determines the subnet address and mask of the primary scope. Use this attribute when multiple logical IP subnets are present on the same physical network.

renew-only [bool](#)

Controls whether to allow existing clients to reacquire their leases, but not offer any leases to new clients. Note that a renew-only scope does not change the client associated with any of its leases (other than to allow a client, currently using what the server believes is an available IP address, to continue using the address).

restrict-to-admin-allocation [bool](#) default = disabled

Controls whether the scope is restricted to administrative requests to allocate the next available address. If set, the server will only respond to a client with an address from this scope if it has been pre-allocated to the client.

restrict-to-reservations [bool](#) default = disabled

Controls whether the scope is restricted to client (or lease) reservations. If enabled, the DHCP server will not automatically assign addresses to clients but instead requires the address to be supplied by a reservation, either a lease reservation or a client reservation, which is specified via a client entry or through an extension and the environment dictionary.

selection-tag-list [nlist](#)(obj(0))

Associates a comma-separated list of selection tags with a scope. The scope compares a client's selection criteria to this list in order to determine whether the client can obtain a lease from the scope.

subnet [subnet](#) required

The network address of the IP subnet that this scope represents.

tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

vpn-id [int](#) default = 0, immutable

Displays the identifier of the DHCP VPN that contains the addresses in this scope. Define this value with the vpn vpn-name create id command. Once set, you cannot change this value.

## scope-policy

scope-policy - Adds DHCP policy information to a scope

### Synopsis

```
scope-policy <name> delete
scope-policy <name> set <attribute>=<value>
                        [<attribute>=<value> ...]
scope-policy <name> get <attribute>
scope-policy <name> unset <attribute>

scope-policy <name> disable <attribute>
scope-policy <name> enable <attribute>
scope-policy <name> show

scope-policy <name> setLeaseTime <time-val>
scope-policy <name> getLeaseTime

scope-policy <name> setOption <opt-name | id> <value>
                        [-blob | -expression] [-roundrobin]
scope-policy <name> getOption <opt-name | id>
scope-policy <name> unsetOption <opt-name | id>
scope-policy <name> listOptions

scope-policy <name> setVendorOption <opt-name | id> <opt-set-name>
                        <value> [-blob | -expression]
scope-policy <name> getVendorOption <opt-name | id> <opt-set-name>
scope-policy <name> unsetVendorOption <opt-name | id> <opt-set-name>
scope-policy <name> listVendorOptions
```

## Description

The `scope-policy` command lets you configure embedded DHCP policy information for a DHCP scope. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case a scope. A scope policy is created implicitly when you first reference it, and is deleted when the scope is deleted.

See the help file for the `policy` command for more information.

## Examples

```
nrcmd> scope-policy scope-example set backup-pct=30
nrcmd> scope-policy scope-example enable allocate-first-available
```

## Status

## See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-template-policy](#)

## Attributes

### `affinity-period` [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the `inhibit-all-renews` attribute or the `inhibit-renews-at-reboot` attribute.

### `allow-client-a-record-update` [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

### `allow-dual-zone-dns-update` [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

### `allow-lease-time-override` [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

### `allow-non-temporary-addresses` [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA NA) addresses. The default is to allow clients to request non-temporary addresses.

### `allow-rapid-commit` [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are checked.

The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses.  
The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it.  
The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603).  
Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified.  
Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64.  
If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebinding operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used.  
The default is the value of the default-prefix-length.  
This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this value, the client is sent this value.  
This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when

it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

#### max-client-rebinding-time [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

#### max-client-renewal-time [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

#### max-leases-per-binding [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

#### offer-timeout [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

#### packet-file-name [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

#### packet-server-name [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

#### packet-siaddr [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

#### permanent-leases [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

#### preferred-lifetime [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, it overrides the preferred-lifetime.

#### reconfigure [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

1 allow	Allows clients to request reconfiguration support and the server will honor the request (default).
2 disallow	Allows clients to request reconfiguration support but the server will not honor the clients' request.
3 require	Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not include a Reconfigure-Accept option.

This attribute has special handling during the policy hierarchy

processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

**reconfigure-via-relay** [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

**reverse-dnsupdate** [nameref\(0\)](#)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

**server-lease-time** [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

**shortest-prefix-length** [rangeint\(0-128\)](#)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

**split-lease-times** [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**unavailable-timeout** [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system default\_policy, then the default is 86400 seconds (or 24 hours).

**use-client-id-for-reservations** [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

**v4-bootp-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

**v4-reply-options** [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

**v6-max-client-preferred-lifetime** [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value. This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server.



The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### v6-max-client-rebinding-time [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### v6-max-client-renewal-time [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### v6-max-client-valid-lifetime [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsoo-allowed-options [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## scope-template

scope-template - Configures a scope template

### Synopsis

```
scope-template list
scope-template listnames
scope-template listbrief

scope-template <name> create [<attribute>=<value> ...]
scope-template <name> delete
scope-template <name> set <attribute>=<value> [<attribute>=<value> ...]
scope-template <name> get <attribute>
scope-template <name> unset <attribute>

scope-template <name> disable <attribute>
scope-template <name> enable <attribute>
scope-template <name> show
```

```

scope-template <name> create clone=<clone-name>
scope-template <name> apply-to <all | <scopel>[,...]>

scope-template < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
scope-template < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
scope-template <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The `scope-template` command lets you configure a template to use when creating scopes.

The `pull`, `push`, and `reclaim` commands are only available when connected to a regional cluster. For `push` and `reclaim`, a list of clusters or "all" may be specified.

## Examples

```

nrcmd> scope-template scope-example create
nrcmd> scope-template scope-example set backup-pct=30
nrcmd> scope-template scope-example enable allocate-first-available

```

## Status

## See Also

[scope](#)

## Attributes

### allocate-first-available [bool](#)

This boolean attribute forces all allocation of new IP addresses from this scope to be made from the first available IP address, rather than the default of the "least recently used" IP address. If this attribute is not set (unset), then the decision on whether to allocate the first available IP address in the scope will be controlled by the DHCP server attribute `priority-address-allocation`. If `priority-address-allocation` is set (and `allocate-first-available` for the scope is not set (unset)), then the scope will allocate addresses as if `allocate-first-available` was set. If `allocate-first-available` has been explicitly configured (either enabled or disabled) for a scope, then for that scope the setting of `priority-address-allocation` has no meaning.

### allocation-priority [int](#)

You can use the `allocation-priority` to assign an ordering to scopes, such that allocation of IP addresses will take place from acceptable scopes with a higher priority until the IP addresses in all those scopes are exhausted. An `allocation-priority` of 0 is treated as not having an `allocation-priority`. You can mix scopes with an `allocation-priority` along with those without an `allocation-priority` (or with an `allocation-priority` of 0, which is the same thing) in the same network. In this case, the scopes with an `allocation-priority` will be examined for acceptability prior to those scopes with no `allocation-priority` (or an `allocation-priority` of 0). If this attribute is not set (unset), then the `allocation-priority` of the scope will be controlled by the DHCP server attribute `priority-address-allocation`. If `priority-address-allocation` is set (and `allocation-priority` for the scope is not set (unset)), then the `allocation-priority` for the scope will be the network number of the scope. If `allocation-priority` has been explicitly configured for a scope, then for that scope the setting of `priority-address-allocation` has no meaning.

### backup-pct [percent](#)

The percentage of available addresses that the main server should send to the backup server. If defined for a scope, it must be defined for the scope in the main server. If it is defined in a backup server, it is ignored (to enable copying of configurations). When defined, values will override the failover pair `backup-pct`. The defined value will be used for this scope, whether or not this scope supports `dynamic-bootp`. If set to zero (0), no addresses will be sent to the backup server. Because zero is a significant value, once set, this attribute must be unset in order for this scope to use the failover pair values for `backup-pct` or `dynamic-bootp-backup-pct`. Note: If the failover pair is configured to use load balancing, the percentage is ignored and 50% is used.

### bootp [bool](#)

Controls whether the server will accept BOOTP requests for this scope. If you want clients to always receive the same addresses, you need to reserve IP addresses for all your BOOTP clients.

**deactivated** [bool](#)

A deactivated scope will not extend leases to any clients. It treats all of the addresses in its ranges as if they were individually deactivated.

**description** [string](#)

Describes the scope template.

**dhcp** [bool](#)

Controls whether the server will accept DHCP requests for this scope. Typically, this is only disabled when bootp is enabled for that scope, to allow only a scope to be used exclusively for BOOTP.

**dns-host-bytes** [rangeint](#)(1-4)

This value tells DHCP how many of the bytes in a lease IP address to use when forming in-addr.arpa names. The server forms names in the in-addr zone by prepending dns-host-bytes of IP address (in reverse order) to the reverse zone name. If this is unset, the server will synthesize an appropriate value based on the scope subnet size.

**dynamic-bootp** [bool](#)

Controls whether the server will accept dynamic BOOTP requests for this scope. Dynamic BOOTP requests are BOOTP requests that do not match a reservation, but could be satisfied from the available lease pool. To use this feature you must also enable bootp.

**embedded-policy** [obj](#)(0)

The embedded policy object for this scope.

**free-address-config** [nameref](#)(0)

The free-address trap configuration to use for this individual scope.

**grace-period** [time](#)

The length of time between the expiration of a lease and the time it is made available for re-assignment. This attribute is set in the scope embedded policy.

**ignore-declines** [bool](#)

This attribute controls whether the DHCP server will process a DHCPDECLINE request referencing an IP address in this scope. If this attribute is enabled, then the DHCP server will ignore all declines which reference an IP address in this scope. If this attribute is not set, the DHCP server will set to UNAVAILABLE every IP address which is referenced in a DHCPDECLINE message. The default value is false, so that DHCPDECLINE messages are processed normally, and the IP addresses referenced in these DHCPDECLINE messages are set to UNAVAILABLE.

**name** [string](#) required,unique

The name of this scope template.

**offer-timeout** [time](#)

If the server offers a lease to a client, but the offer is not accepted, the server will wait the specified number of seconds before making the lease 'available' again. This attribute is set in the scope embedded policy.

**options-expr** [expr](#)

An expression to define the list of embedded policy options to be created for a scope object.

**ping-clients** [bool](#)

Controls whether the server should attempt to ping addresses before offering leases.

**ping-timeout** [int](#)

The number of milliseconds the DHCP server should wait for ping responses. If you make this value too large, you will slow down the lease offering processes. If you make this value too small, you will reduce the effectiveness of pinging addresses before offering them.

**policy** [nameref](#)(0) default = default

The name of the policy associated with this scope.

### ranges-expr [expr](#)

An expression to define the list of scope ranges to be created for a scope object.

### renew-only [bool](#)

Controls whether to allow existing clients to reacquire their leases, but not offer any leases to new clients. Note that a 'renew-only' scope will not change the client associated with any of its leases (other than to allow a client currently using what the server believes is an available IP address to continue using it).

### restrict-to-reservations [bool](#) default = disabled

Controls whether the scope is restricted to client (or lease) reservations. If enabled, the DHCP server will not automatically assign addresses to clients but instead requires the address to be supplied by a reservation, either a lease reservation or a client reservation, which is specified via a client entry or through an extension and the environment dictionary.

### router-host [int](#) default = 1

Defines the address offset for the giaddr address on the subnet. It is used to create the router interface address, which is an AT IPNET address that combines the giaddr and scope subnet, when Push Subnet is used to create both a scope and a router interface from the scope template.

### scope-description-expr [expr](#)

Defines an AT STRING expression to apply to the description on the scope object created when using the template.

### scope-name [expr](#)

An expression to define the name of the scope object created when using the scope template.

### selection-tag-list [nlist](#)(obj(0))

The list of selection tags to associate with a scope.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

### update-dns-for-bootp [bool](#)

If the server is replying to a BOOTP request, and is offering a lease from a Scope which is configured to perform DNS updates, it will check this property before beginning the DNS update. This feature allows an administrator to prevent DNS updates for BOOTP clients, while allowing updates for DHCP clients. 'Update-dns-for-bootp' can be controlled globally with the 'server' command; that global command can be overridden by individual Scopes as necessary.

## scope-template-policy

scope-template-policy - Edits the DHCP policy embedded in a scope-template

### Synopsis

```
scope-template-policy <name> delete
scope-template-policy <name> set <attribute>=<value>
                             [<attribute>=<value> ...]
scope-template-policy <name> get <attribute>
scope-template-policy <name> disable <attribute>
scope-template-policy <name> enable <attribute>
scope-template-policy <name> show

scope-template-policy <name> setLeaseTime <time-val>
scope-template-policy <name> getLeaseTime

scope-template-policy <name> setOption <opt-name | id> <value>
                             [-blob | -expression] [-roundrobin]
scope-template-policy <name> getOption <opt-name | id>
scope-template-policy <name> unsetOption <opt-name | id>
scope-template-policy <name> listOptions

scope-template-policy <name> setVendorOption <opt-name | id>
                             <opt-set-name> <value> [-blob | -expression]
scope-template-policy <name> getVendorOption <opt-name | id>
                             <opt-set-name>
scope-template-policy <name> unsetVendorOption <opt-name | id>
                             <opt-set-name>
```

## scope-template-policy <name> listVendorOptions

### Description

The scope-template-policy command lets you configure a DHCP policy embedded in a DHCP scope template. An embedded policy is a collection of DHCP option values and settings associated with (and named by) another object -- in this case a scope template. A scope-template-policy is created implicitly when you first reference it, and is deleted when the scope-template is deleted.

See the help file for the policy command for more information.

### Examples

```
nrcmd> scope-template-policy exampleScope set default-prefix-length=32
nrcmd> scope-template-policy exampleScope enable allow-rapid-commit
```

### Status

### See Also

[policy](#), [client-policy](#), [client-class-policy](#), [dhcp-address-block-policy](#), [link-policy](#), [link-template-policy](#), [prefix-policy](#), [prefix-template-policy](#), [scope-policy](#)

### Attributes

#### affinity-period [time](#)

Associates a lease in the AVAILABLE state with the client that last held the lease. If the client requests a lease during the affinity period, it is granted the same lease; that is, unless renewals are prohibited, then it is explicitly not given the lease. Because of the vast IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs reassignment to a different client, and there is no reason to hold on to this information for that long. To prohibit renewals enable either the inhibit-all-renews attribute or the inhibit-renews-at-reboot attribute.

#### allow-client-a-record-update [bool](#) default = disabled

Determines if a client is allowed to update A records. If the client sets the flags in the FQDN option to indicate that it wants to do the A record update in the request, and if this value is TRUE, the server allows the client to do the A record update; otherwise, based on other server configurations, the server does the A record update.

#### allow-dual-zone-dns-update [bool](#) default = disabled

Enables DHCP clients to perform DNS updates into two DNS zones. To support these clients, you can configure the DHCP server to allow the client to perform an update, but also to perform a DNS update on the client's behalf.

#### allow-lease-time-override [bool](#) default = disabled

Gives the server control over the lease period. Although a client can request a specific lease time, the server need not honor the request if this attribute is set to false (the default). Even if set to true, clients can request only lease times that are shorter than those configured for the server.

#### allow-non-temporary-addresses [bool](#) default = true

Determines whether DHCPv6 clients can request non-temporary (IA\_NA) addresses. The default is to allow clients to request non-temporary addresses.

#### allow-rapid-commit [bool](#) default = false

Determines whether DHCP clients can use a Discover/Solicit with the Rapid Commit option to obtain configuration information with fewer messages. To permit this, make sure that a single DHCP server is servicing clients. For DHCPv6, this attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked:

- If any of the prefix policies has this attribute set to FALSE, Rapid Commit is not allowed.
- If at least one has it set to TRUE, Rapid Commit is allowed.
- Otherwise, the remaining policies in the hierarchy are

checked.  
The default is not to allow clients to use Rapid Commit.

**allow-temporary-addresses** [bool](#) default = true

Determines whether DHCPv6 clients can request temporary (IA\_TA) addresses.  
The default is to allow clients to request temporary addresses.

**default-prefix-length** [rangeint](#)(0-128) default = 64

For delegation, specifies the default length of the delegated prefix, if a router (client) does not explicitly request it.  
The default length must always be greater than or equal to the prefix length of the prefix range.

**excluded-prefix** [prefix](#)

Specifies the excluded prefix (bits and prefix-length) to be sent to DHCPv6 clients that requested the PD Exclude Option (see RFC 6603).  
Note that only the bits between the actual delegated prefix's prefix length and the excluded-prefix's prefix length are used from the prefix specified.  
Using the example in RFC 6603, this attribute would be set to 0:0:0:f::/64.  
If configured, the server will determine the excluded prefix when a lease is assigned to a client; but it is only sent to the client if the client requests the PD Exclude option.

**forward-dnsupdate** [nameref](#)(0)

Specifies the name of the update configuration that determines which forward zones to include in updates.

**forward-zone-name** [dname](#)

Designates an optional forward zone for DNS updates that overrides the forward-zone-name configured in the forward (or reverse) DNS Update Configuration object.  
Note that a forward zone name is required for any DNS updates to occur (whether forward, reverse, or both).

**giaddr-as-server-id** [bool](#) default = false

Enables the DHCP server to set the server-id option on a DHCP OFFER and a DHCP ACK to the giaddr of the incoming packet, instead of the IP address of the server (the default action).  
This causes all unicast renews to be sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option-82 information appended to the packet.  
Some relay agents may not support this capability and, in some complex configurations, the giaddr might not actually be an address to which the DHCP client can send a unicast packet. In these cases, the DHCP client cannot renew a lease, and must always perform a rebind operation (where the DHCP client broadcasts a request instead of sending a unicast to what it believes is the DHCP server).

**grace-period** [time](#) default = 5m

Defines the length of time between the expiration of a lease and the time it is made available for reassignment.

**inhibit-all-renews** [bool](#) default = false

Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server.

**inhibit-renews-at-reboot** [bool](#) default = false

Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

**lease-retention-limit** [bool](#) default = disabled

If enabled and the DHCP server's lease-retention-max-age is configured to a non-zero value, times in leases subject to this policy will not be allowed to grow older than lease-retention-max-age. As they progress toward lease-retention-max-age, they will periodically be reset to lease-retention-min-age in the past.

**limitation-count** [int](#)

Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active and valid leases.

**longest-prefix-length** [rangeint](#)(0-128)

For prefix delegation, specifies the longest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is longer than this, this length is used.  
The default is the value of the default-prefix-length.  
This prefix length must always be greater than or equal to the prefix length of the prefix range.

**max-client-lease-time** [rangetime](#)(60s-2y)

Specifies the maximum client lease time that the server is allowed to send to the client. If the calculated lease time is greater than this

value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server, including when it expects the client to renew (T1). The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the max-client-renewal-time and max-client-rebinding-time attributes.

**max-client-rebinding-time** [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-lease-time as T2 must be less than or equal to the lease time.

**max-client-renewal-time** [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the max-client-rebinding-time or max-client-lease-time as T1 must be less than or equal to T2 (and the lease time).

**max-leases-per-binding** [rangeint](#)(0-65535)

Specifies the maximum number of leases that a client may use per binding from an allocation group. This applies to DHCPv6 only. Explicit or implicit allocation groups only limit new server initiated allocations to a binding. They do not limit the overall leases a client may use. Leases may have been assigned because of differences in the configuration, reservations, communication with the failover partner, client requests, or from using extensions to alter lease acceptability (lease acceptability extensions can still override the limits as well). This attribute can be used to limit the number of leases. The server only applies a configured limit for client Solicit, Request, Renew, and Rebind requests and the server will prefer the leases that were most recently provided to the client. However, when leases have the same time, the result will be random as to which lease(s) will be revoked.

**offer-timeout** [time](#) default = 2m

Instructs the server to wait a specified amount of time when it has offered a lease to a client, but the offer is not yet accepted. At the end of the specified time interval, the server makes the lease available again.

**packet-file-name** [string](#)

Identifies the boot-file to use in the boot process of a client. The server returns this file name in the 'file' field of its replies. The packet-file-name cannot be longer than 128 characters.

**packet-server-name** [string](#)

Identifies the host-name of the server to use in a client's boot process. The server returns this file name in the 'sname' field of its replies. The packet-server-name field cannot be longer than 64 characters.

**packet-siaddr** [ipaddr](#)

Identifies the IP address of the next server in the client boot process. For example, this might be the address of a TFTP server used by BOOTP clients. The server returns this address in the 'siaddr' field of its replies.

**permanent-leases** [bool](#) default = disabled

Indicates whether leases using this policy are permanently granted to requesting clients. If leases are permanently granted, the dhcp-lease-time will be infinite.

**preferred-lifetime** [time](#) default = 1w

Assigns the default and maximum preferred lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that the address is preferred; that is, its use is unrestricted. When the preferred lifetime expires, the address becomes deprecated and its use is restricted. Note: For IA TA's, if the (expert mode) min-preferred-lifetime is configured, It overrides the preferred-lifetime.

**reconfigure** [enumint](#)(allow=1, disallow=2, require=3) default = allow

Controls DHCPv6 client reconfiguration support:

1 allow	Allows clients to request reconfiguration support and the server will honor the request (default).
2 disallow	Allows clients to request reconfiguration support but the server will not honor the clients' request.
3 require	Requires clients to request reconfiguration support and the server drops client Solicit and Request messages that do not

include a Reconfigure-Accept option. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked as follows:

- If any of the prefix policies has this attribute set to disallow or require, that setting is used.
- Otherwise, if at least one has it set to allow, Reconfigure is allowed.
- If no prefix policies have this attribute set, the remaining policies in the hierarchy are checked.

#### reconfigure-via-relay [bool](#) default = false

Controls whether the server should prefer unicasting or relaying DHCPv6 Reconfigure messages. If false (the default), the server prefers to unicast Reconfigure messages if the client has one or more valid statefully assigned addresses. If true, the server prefers to send Reconfigure messages via the relay agent unless no relay agent information is available.

Note: When you use this attribute, consider that:

- In networks where the DHCPv6 server cannot communicate directly with its client devices, for example, where firewalls or VPNs are in use, set this value to true.
- The DHCPv6 server does not use embedded and named policies configured on a client when it evaluates this attribute.
- The relay agent cannot be used if the Relay-Forw message came from a link-local address.

#### reverse-dnsupdate [nameref\(0\)](#)

Specifies the name of the update configuration that determines which reverse zones to include in a DNS update.

#### server-lease-time [time](#)

Tells the server how long a lease is valid. For more frequent communication with a client, you might have the server consider leases as leased for a longer period than the client considers them. This also provides more lease-time stability. This value is not used unless it is longer than the lease time in the dhcp-lease-time option found through the normal traversal of policies.

#### shortest-prefix-length [rangeint\(0-128\)](#)

For prefix delegation, specifies the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, this length is used. The default is the value of the default-prefix-length. This prefix length must always be greater than or equal to the prefix length of the prefix range.

#### split-lease-times [bool](#) default = disabled

Specifies a value that the DHCP server might use internally to affect lease times. If enabled, the DHCP server still offers clients lease times that reflect the configured lease-time option from the appropriate policy; but the server bases its decisions regarding expiration on the 'server-lease-time' value.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### unavailable-timeout [time](#) default = 24h

Permits the server to make a lease unavailable for the time specified and then to return the lease to available state. If there is no value configured in the system default\_policy, then the default is 86400 seconds (or 24 hours).

#### use-client-id-for-reservations [bool](#) default = off

Controls how the server database checks for reserved IP addresses. By default, the server uses the MAC address of the DHCP client as the key for its database lookup. If this attribute is set to true (enabled), then the server does the check for reserved addresses using the DHCP client-id, which the client usually sends. In cases where the DHCP client does not supply the client-id, the server synthesizes it, and uses that value.

#### v4-bootp-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all BOOTP clients.

#### v4-reply-options [nlist\(obj\(0\)\)](#)

Lists the options the server returns to all DHCPv4 clients, whether or not the client specifically asks for the option data.

#### v6-max-client-preferred-lifetime [rangetime\(60s-2y\)](#)

Specifies the maximum client preferred lifetime that the server is allowed to send to the client. If the calculated preferred lifetime is greater than this value, the client is sent this value.



This attribute limits the preferred lifetime sent to the client, it does not impact any of the time calculations made by the server. The renewal (T1) and rebinding (T2) times given to the client will be based on the lease time actually sent to the client and may further be limited by the v6-max-client-renewal-time and v6-max-client-rebinding-time attributes.

#### v6-max-client-rebinding-time [rangetime](#)(30s-2y)

Specifies the maximum client rebinding time (T2) that the server is allowed to send to the client. If the rebinding time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-preferred-lifetime as T2 must be less than or equal to the preferred lifetime.

#### v6-max-client-renewal-time [rangetime](#)(30s-2y)

Specifies the maximum client renewal time (T1) that the server is allowed to send to the client. If the renewal time is greater than this value, the client is sent this value. This attribute limits the time sent to the client, it does not impact any of the lease time calculations made by the server. The actual value sent to the client may be further limited by the v6-max-client-rebinding-time or v6-max-client-preferred-lifetime as T1 must be less than or equal to T2 (and the preferred lifetime).

#### v6-max-client-valid-lifetime [rangetime](#)(60s-2y)

Specifies the maximum client valid lifetime that the server is allowed to send to the client. If the calculated valid lifetime is greater than this value, the client is sent this value. This attribute limits the valid lifetime sent to the client, it does not impact any of the time calculations made by the server.

#### v6-reply-options [nlist](#)(obj(0))

Lists the options that should be returned in any replies to DHCPv6 clients. This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### v6-rsoo-allowed-options [nlist](#)(obj(0))

Lists the RSOO-enabled options that server will process, if provided by a relay and no option is otherwise available. See RFC 6422 for more details on RSOO (Relay Supplied Options Option). This attribute has special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link. The Prefixes for the Link are processed in alphabetic (case blind) order. For each Prefix, the embedded and then named policy are checked. Only Prefixes to which the client has access (based on selection tags, etc.) are checked.

#### valid-lifetime [time](#) default = 2w

Assigns the default and maximum valid lifetime for leases to DHCPv6 client interfaces. Expressed in seconds and relative to the time the server sent the packet, this attribute sets the length of time that an address remains valid. When this period of time expires, the address becomes invalid and unusable. The valid lifetime must be greater than or equal to the preferred lifetime. Note: For IA TA's, if the (expert mode) min-valid-lifetime is configured, it overrides the valid-lifetime.

#### view-id [int](#)

Designates the optional view associated with zones used for DNS update that overrides the view-id configuration in forward (reverse) DNS Update configuration object.

## server

server - Configures and controls the server objects

### Synopsis

The server keyword is optional. You can enter all commands starting with just the server type (<server>).

```
server <server> disable <attribute>
server <server> enable <attribute>
```

```
server <server> get <attribute>
```

```
server <server> start
server <server> stop
server <server> reload
```

```
server <server> getHealth
server <server> getStats

server <server> setDebug <category-list>=<value>
server <server> unsetDebug
server <server> getDebug

server <server> serverLogs show
server <server> serverLogs nlogs=<nlogs> logsize=<logsize>
```

## Description

The server command is used to send general control messages and queries to the servers in the cluster. <server> may be ccm, cdns, dhcp, dns, snmp, or tftp.

Not all servers support all commands.

The stop command stops the specified server.

The start command starts the specified server.

The reload command reloads the specified server, possibly with new configuration information.

The getHealth command gets the specified server's current health value.

The getStats command gets the specified server's current statistics.

The setDebug command sets the debugging level. The debugging information is written to the server's log file.

The unsetDebug command turns off debugging output.

The serverLogs show command displays the number of log files and the maximum size for each file.

The serverLogs command allows setting the two server logging parameters, nlogs and logsize. Either or both may be specified in the command, and changes occur only to the one(s) specified. When setting logsize, the value may be suffixed with K or M to signify units of thousands or millions. Note that in order for these changes to take effect you must save the changes and restart the server agent.

```
server dhcp serverLogs nlogs=6 logsize=500K
server dns serverLogs logsize=5M
```

The features that can be enabled or disabled are:

start-on-reboot

For any server, controls whether this server is started by the Network Registrar server agent. Disabling this feature for a server is useful for clusters that want to provide a single protocol service. By default both the DNS and DHCP services are enabled.

The properties that can be read with the get command are:

version

The version string for this server.

## Examples

```
nrcmd> server DNS stop
nrcmd> server DNS start
nrcmd> server DHCP getHealth
nrcmd> server DNS disable start-on-reboot
```

## See Also

[ccm](#), [cdns](#), [dhcp](#), [dns](#), [snmp](#), [tftp](#)

## session

session - Configures nrcmd program session parameters

## Synopsis

```

session set <attribute>=<value> [-save]
session get <attribute>
session enable <attribute> [-save]
session disable <attribute> [-save]
session unset <attribute> [-save]
session show

session cache <refresh | clear>

session listNetInterfaces

session listProcesses

session log [<filename>]

session listbrief set <class> | <command> [<format>]
session listbrief unset <class> | <command>
session listbrief show [<class> | <command> | all]

```

## Description

The `session` command lets you view and set session parameters, such as the session visibility and the default output format of your nrcmd program session. If `-save` is specified when changing a parameter that can be saved, and you are connected to a cluster that supports this feature, the setting is saved in your user preferences. Future nrcmd sessions to that cluster will use the saved settings.

The `listNetInterfaces` command returns a list of the network interfaces that are present on the machine running Network Registrar. Both IPv4 and IPv6 interfaces are included.

The `listProcesses` command returns a list of the Network Registrar processes running on the cluster, provided statistics history is enabled.

The `log` command closes the currently open log file, if any, and opens a new log file to which subsequent output is written if a filename is specified.

The `session listbrief` commands are used to manage the user's custom definitions of formats for subsequent `listbrief` operations for that object class or nrcmd command. The default formats, and details on the format specifications, can be found in the `conf/nrcmd-listbrief-defaults.conf` file. System-wide definitions can be added to the `conf/nrcmd-listbrief-custom.conf` file. The user's definitions are saved across sessions.

```

session listbrief show
  Displays the user's customized definitions.
session listbrief show <class>|<command>
  Displays the definition for the object class or command
  (whether user or system defined).
session listbrief show all
  Displays all definitions.
session listbrief set <class>|<command>
  Resets the display format to the default for the class
  or command.

```

## Features

`admin-edit-mode` (regional only)  
 The edit mode currently in effect when editing admins at a regional cluster. The valid values are 'staged' and 'synchronous' (or 'sync'). The value 'default' will use the value configured at the CCM server.

`cache`  
 The CLI caches many configuration objects that it reads. If multiple users are making changes simultaneously, one CLI instance may have cached an out-of-date version of an object. The cache refresh command causes the CLI to clear its local cache of all unmodified objects, forcing the CLI to re-read objects from the configuration database. The cache clear command forces the CLI to clear all cached data, whether or not unsaved changes have been made.

#### cluster

The name of the current cluster. This is a read-only property.

#### current-view (or view)

The name of the current dns-view presently in effect. This is a read/write property. This is the DNS view used in commands which need a view and when no view is explicitly specified when the command is issued. The reserved view "Default" is the view which contains all of the zones and resource records not in any explicitly named view, and is the default view for the session.

Note: If the session is interactive, the "nrcmd>" prompt is changed to "nrcmd [View:<view-name>]>" except when the view-name is "Default".

#### current-vpn (or current-namespace or vpn)

The name of the current VPN presently in effect. This is a read/write property. This is the VPN used in commands which need a VPN (e.g., lease, report, export) and no VPN is explicitly specified when the command is issued. The reserved vpn-name "global" is the VPN which contains all of the IP addresses not in any explicitly named VPN, and is the default current-vpn for the session.

Note: If the session is interactive, the "nrcmd>" prompt is changed to "nrcmd [VPN:<vpn-name>]>" except when the vpn-name is global.

#### default-format

The default-format is an enumerated string that can have the following settings. If not set, it is interpreted as script. Defaults to user for interactive sessions.

user - Show objects in user-friendly form: one property per line.

script - Show objects in script friendly form: one object per line.

#### dhcp-edit-mode (formerly scope-edit-mode)

The edit mode currently in effect when editing DHCP scopes and reservations. The valid values are 'staged' and 'synchronous' (or 'sync'). The value 'default' will use the value configured at the CCM server.

#### dns-edit-mode (formerly zone-edit-mode)

The edit mode currently in effect when editing DNS zones. The valid values are 'staged' and 'synchronous' (or 'sync'). The value 'default' will use the value configured at the CCM server.

#### error

The error detail for the last request to the CCM server that failed. This report is often used to get more detailed information about why a nrcmd command failed to complete successfully.

Note: The error detail may not reflect the final result of the nrcmd command because some commands result in multiple requests to CCM, one or more of which may have failed and the error detail reports only on the last failed operation (which may not be the last operation).

#### groups

Displays the list of groups associated with the current user.

#### roles

Displays the list of roles associated with the current user.

#### output-using

Specifies the program used to display nrcmd output. Normally output is written to directly to the screen. However, if this command is used to specify a program, that program is run with the output in a temporary file and can be used to display the output. Typically this would be used to specify the more (or less) program. This setting may be saved for future sessions (if -save is specified when changed).

#### show-unset

Controls whether unset properties are included when showing objects. If unset, the default-format determines whether these properties are included (user includes them, script does not). This property can be enabled (to include unset properties), disabled (to exclude unset properties), or unset (to use the default-format setting). This setting may be saved for future sessions (if -save is specified when changed).

**tenant**  
Gets to sets the tenant for the current session. All subsequent operations are done under that tenant.

**user-name**  
The name of the current user. This is a read-only property.

**version**  
Retrieves the version associated with the Network Registrar cluster.

**visibility**  
The session visibility setting. The default level is 5, which is the normal operating mode. Visibility level 3 can be set to view 'expert' level properties. These additional, reserved properties should be modified with caution. Incorrect settings for these properties can result in network outages. The default settings are the best choice in virtually all cases and should not be changed, except to address a specific network issue. Changes should not be attempted without assistance from the Cisco TAC. This setting may be saved for future sessions (if -save is specified when changed).

## Examples

## Status

## See Also

# show

show - Views current configuration or status

## Synopsis

```
show license status
show license summary
show license reservation
show license UDI
show license usage
show license eventlog <days>
show license all
show license tech-support

show call-home detail
```

## Description

These commands are available in smart license configuration mode.

The **show license** command allows you to view current smart license configuration, usage or status.

The **show call-home detail** command allows you to view current call home configuration.

## Examples

## Status

## See Also

[smart](#), [call-home](#)

## smart

smart - Views and updates smart license information

### Synopsis

smart

```
[no] license smart enable
[no] license smart reservation
[no] license smart custom_id <id>
[no] license smart transport [callhome|smart]
[no] license smart url [default|<url>]
[no] license smart http-proxy <addr> port <port-number>
[no] license smart privacy <all|hostname|version>
[no] debug smart_lic <error|trace|debug|all>

license smart register idtoken <token> [force]
license smart deregister
license smart renew ID
license smart renew auth
license smart reservation request <local|all>
license smart reservation install <auth-code>
license smart reservation install file <file-path>
license smart reservation return <local|all>
license smart reservation return authorization <auth-code>
license smart reservation cancel <local|all>
license smart export request <local|all> <feature-name>
license smart export return <local|all> <feature-name>
license smart export cancel
license smart factory reset
license smart clear eventlog

show license status
show license summary
show license reservation
show license UDI
show license usage
show license eventlog <days>
show license all
show license tech-support
```

### Description

The **smart** command can be used to enter into smart license configuration mode. All smart license and call home commands can only be executed in this mode.

Use **exit** command to return to global configuration mode.

The **license smart** commands allows you to configure smart licenses on a regional cluster. **show license** commands can be used to view current smart license configuration, status and usage. Any smart license configuration command, when executed with 'no' as prefix, unsets the specific configuration.

### Examples

### Status

### See Also

[license](#), [call-home](#), [show](#)

## snmp

snmp - Configures and controls the SNMP server

## Synopsis

```
snmp disable <attribute>
snmp enable <attribute>

snmp get <attribute>
snmp set <attribute>=<value> [<attribute>=<value> ...]
snmp unset <attribute>
snmp [show]
```

## Description

The snmp command configures the Network Registrar SNMP server. The SNMP server makes some Network Registrar statistics available to SNMP clients, and generates SNMP traps based on the status of the other Network Registrar servers.

## Examples

## Status

## See Also

[server](#)

## Attributes

**cache-ttl** [time](#) default = 60s

Controls how long CNR data can be cached by the SNMP server as it responds to SNMP GETs.

**community** [string](#) default = public

SNMP requests must present a matching community-string in order to be processed by the SNMP server.

**local-proxy-only** [bool](#) default = disabled

Defines whether the server accepts queries only from local and proxied sources, or from any source. When using SNMPv3, enabling this is recommended. Enabling this setting overrides any SNMP interface configuration.

**log-settings** [flags](#)(default=1, no-success-messages=2, incoming-packet-detail=3, outgoing-packet-detail=5, scp-detail=6, snmp-detail=7) default = default

Configures log messages. The server logs more or fewer messages depending on this parameter.

**server-active** [bool](#) default = true

If 'true', the server will run when it is started. If 'false', the server will not run when it is started.

**snmp-port** [rangeint](#)(1-65535) default = 4444

The port to listen on for incoming SNMP queries.

**trap-source-addr** [ipaddr](#)

An optional address to use as the sender address in outgoing SNMP trap packets.

**trap-source-ip6address** [ip6addr](#)

An optional IPv6 address to use as the sender address in outgoing SNMP trap packets.

## snmp-interface

snmp-interface - Configures the SNMP server's network interfaces

## Synopsis

```
snmp-interface <name> create [<attribute>=<value>] [<attribute>=<value> ...]
```

```

snmp-interface <name> delete
snmp-interface list
snmp-interface listnames
snmp-interface listbrief
snmp-interface <name> show
snmp-interface <name> set <attribute>=<value> [<attribute>=<value> ...]
snmp-interface <name> get <attribute>
snmp-interface <name> unset <attribute>

snmp-interface <name> enable <attribute>
snmp-interface <name> disable <attribute>

```

## Description

The `snmp-interface` command configures network interfaces for use by the NetworkRegistrar SNMP server. If there are no defined interfaces, the server discovers and uses all available interfaces on the system. When this list is present, the server only uses available interfaces, if any, that match this list.

## Examples

```

nrcmd> snmp-interface example1 create
nrcmd> snmp-interface example1 set address=192.168.0.58
nrcmd> snmp-interface example1 set ip6address=2001:4f8:ffff:0:8125:ef1b:bdc4:4b4e

```

## Status

## See Also

### Attributes

**address** [subnet](#)

The IP address and subnet mask of an interface that the SNMP server should use.

**ip6address** [prefix](#)

The IPv6 address and prefix length of an interface that the SNMP server should use.

**name** [string](#) required,unique

Specifies the user-assigned name of the SNMP server interface.

**snmp-port** [rangeint](#)(1-65535) default = 4444

The port to listen on for incoming SNMP queries.

## subnet

`subnet` - Describes a contiguous range of IP address space in the address-space model

## Synopsis

```

subnet [<vpn-name>/]<address/mask> create
subnet [<vpn-name>/]<address/mask> delete
subnet list
subnet listnames
subnet listbrief
subnet <address> show

subnet <address> get <attribute>
subnet <address> set <attribute>=<value> [<attribute>=<value> ...]
subnet <address> unset <attribute>

subnet <name> push <cluster/failover-pair> [-template=<template-name>]
subnet <name> reclaim [-force]

```



## Description

A subnet is a correctly aligned, power-of-2 sized sequence of IP addresses. In the address space model, subnets are leaf nodes while address-blocks are root, internal or leaf nodes of a tree describing the address space.

The push and reclaim commands are only available when connected to a regional cluster.

## Examples

```
nrcmd> subnet 192.168.0.0 create
nrcmd> subnet 192.168.0.64 set primary-subnet=192.168.0.0
nrcmd> subnet 192.168.0.64 show primary-subnet
```

## Status

## See Also

[address-block](#)

## Attributes

**address** [subnet](#) required,immutable

Sets the subnet address for this subnet. Stored as a network address and a mask determining how large the range is.

**description** [string](#)

Provides a description of this subnet.

**dns-host-bytes** [rangeint](#)(1-4)

Determines the correct in-addr.arpa name to create, when creating reverse zones from a subnet. Based on this value and the subnet size, either a new reverse zone is created, or delegation records are put into the parent reverse zone. If unset, the server synthesizes an appropriate value based on the subnet size. This value parallels the dns-host-bytes attribute in the Scope class.

**failoverpair** [oid](#)

For DHCP allocation, assigns a subnet to a CCMFailoverPair or to a single CCMLcluster object.

**forward-zone-name** [dname](#)

Names the forward zone associated with this subnet.

**interface** [oid](#)

Assigns a subnet to a router interface.

**owner** [oid](#)

Identifies the owner of this object, referenced by OID. Use the owner field to group similarly owned objects and to limit access to objects by their owner. If this attribute is unset, the object is assumed to have the same owner as its parent.

**parent** [oid](#)

Identifies the parent address block for this subnet.

**primary-subnet** [subnet](#)

Identifies the primary subnet number for this subnet. Use this attribute when multiple logical IP subnets are present on the same physical network.

**region** [oid](#)

Associates an object with a region. The object is referenced by OID. Use the region field to group similarly located objects to limit access to objects by their region. If this attribute is unset, the object is assumed to have the same region as its parent.

**reverse-zone-name** [dname](#)

Names the reverse zone associated with this subnet.

**scope-list** [nlist](#)(obj(0)) transient

Provides the list of scope objects associated with this subnet.

**subnet-state** [flags](#)(available=0, ric-allocated=1, dhcp-allocated=2, reclaiming=3) default = available

Shows the current state of this subnet.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**type** [nameref](#)(0)

Identifies the type of this subnet. This attribute contains the name of a CCMAddrSpaceType object that contains more information about the type. If this attribute is unset, the object is assumed to have the same type as its parent.

**vpn-id** [int](#) default = 0, immutable

Specifies the VPN that contains the subnet address for this subnet.

## sync-from-dns

sync-from-dns - Synchronizes CCM from DNS

### Synopsis

**sync-from-dns**

### Description

The sync-from-dns command rebuilds CCM management databases from the DNS server. Live server RR data may have the potential, over time, to go out of synchronization with the managed host data, particularly if live RRs are managed both locally and regionally. The sync-from-dns command can be used to perform synchronization of DNS data from the DNS server(s) to the CCM management database(s).

It is only available when NRCMD is at visibility 3.

### Examples

### Status

### See Also

[session](#)

## task

task - Configures a scheduled task

### Synopsis

```
task <name> create <task-type> <interval> [<sync-obj>]
                    [<attribute>=<value>]
task <name> delete
task list
task listnames
task listbrief
task <name> show
task <name> set <attribute>=<value> [<attribute>=<value> ...]
task <name> get <attribute>
task <name> unset <attribute>
```

```
task <name> enable <attribute>
task <name> disable <attribute>

task <name> run
```

## Description

The task command configures scheduled task objects. These objects can perform periodic operations automatically.

```
task <name> run
  Runs the specified task now.
```

## Examples

## Status

## See Also

### Attributes

**description** [string](#)

Provides a description for this task.

**enabled** [bool](#) default = true

Indicates whether a task is enabled to run. If false, the task will not run. This attribute may be used to temporarily disable a task, without deleting it.

**last-error-list** [nlist\(obj\(0\)\)](#)

Lists the errors recorded if the most recent task execution failed.

**last-status** [retcode](#)

Provides the most recent status recorded when executing this task.

**name** [string](#) required,unique

Names the task.

**next-run-time** [date](#) transient

Indicates the time at which this task is next scheduled to run.

**schedule-interval** [rangetime](#)(15m-4w2d) required

Specifies how often this task should be performed.

**schedule-offset** [rangetime](#)(0-24h)

Provides a fixed time of day for this task. This time is interpreted as a time of day offset from 12 midnight, provided schedule-interval is less than 24 hours, and the offset value is less than the interval. If the offset value is greater than the interval, or the interval is greater than 24 hours, the offset will be ignored. The scheduler for the task will ensure that the first scheduled task occurs at the offset time. For example, if you set the interval to 4 hours and the offset to 2am, the task would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Note you cannot specify a start time of midnight.

**sync-mode** [enumint](#)(exact=1, update=2, complete=3)

Specifies the synchronization mode for the HA DNS pair, DHCP failover pair, or zone distribution map with this task. This attribute is only relevant for task-types sync-dhcp-pair, sync-dns-pair, sync-zd-map, or smart-sync-dhcp-pair.

**sync-object-oid** [oid](#)

Associates an object identifier for the HA DNS pair, DHCP failover pair, zone distribution map, or DNS update map with this task. This attribute is only relevant for task-types sync-dhcp-pair, sync-dns-pair, sync-zd-map,

sync-dns-update-map, or smart-sync-dhcp-pair.

**task-type** [enumint](#)(dhcp-reload=1, dns-reload=2, sync-dhcp-pair=3, sync-dns-pair=4, sync-zd-map=5, sync-dns-update-map=6, cdns-reload=7, smart-sync-dhcp-pair=8) required

Controls the type of task to be scheduled:

1. dhcp-reload
  - a. Reload the local DHCP server.
2. dns-reload
  - a. Reload the local DNS server.
3. sync-dhcp-pair
  - a. Reload main DHCP server.
  - b. If reload successful, sync failover configuration from main to backup.
  - c. If sync successful, reload backup DHCP server.
4. sync-dns-pair
  - a. Reload main DNS server.
  - b. If reload successful, sync HA pair configuration from main to backup.
  - c. If sync successful, reload backup DNS server.
5. sync-zd-map
  - a. Reload primary DNS server, or HA main server.
  - b. If reload successful, sync zone distribution map.
  - c. If sync successful, reload HA backup DNS server (if configured).
  - d. If reload successful, reload secondary server(s).
6. sync-dns-update-map
  - a. Sync dns update map to DHCP and DNS servers.
  - b. If sync successful, reload local and remote server(s).
7. cdns-reload
  - a. Reload the local CDNS server.
8. smart-sync-dhcp-pair
  - a. Reload main DHCP server if there are any DHCP configuration updates made since the last time the server read the full configuration.
  - b. If reload done and fails, abort the task.
  - c. Sync configuration from main to backup.
  - d. If synchronization fails, abort the task.
  - e. Reload the backup server if there are any DHCP configuration updates on the backup since the last time the backup server read the full configuration.

## tenant

tenant - Configures a tenant

### Synopsis

```
tenant <tag> create <tenant-id> [<attribute>=<value>]
tenant <tag> delete
tenant list
tenant listnames
tenant listbrief
tenant <tag> show
tenant <tag> set <attribute>=<value> [<attribute>=<value> ...]
tenant <tag> get <attribute>
tenant <tag> unset <attribute>

tenant <tag> enable <attribute>
tenant <tag> disable <attribute>

tenant <tag> | all > pull <ensure | replace | exact >
    <cluster-name> [-report-only | -report]
tenant <tag> | all > push <ensure | replace | exact >
    <cluster-list> [-report-only | -report]
tenant <tag> reclaim <cluster-list> [-report-only | -report]
```

### Description

The tenant command configures objects representing a tenant. Tenants may be used to segment data stored on both regional and local clusters. They are intended for managed services providers, giving them a central management point for their customer configurations and network data, or cloud service providers, where many small customer configurations may be consolidated on a set of infrastructure servers. Any given local cluster may be associated with one or more tenants, but within a local cluster, the address pools and domain names assigned to a given tenant must be non-overlapping.

WARNING: Deleting a tenant also deletes all of the tenant's configuration objects.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

### Examples

## Status

## See Also

### Attributes

**description** [string](#)

Provides additional descriptive information for this tenant.

**name** [string](#)

Provides a descriptive name for this tenant.

**tag** [tag](#) required,unique

Specifies the unique tag for this tenant.

**tenant-id** [rangeshort](#)(1-65535) required,unique,immutable

Specifies the unique identifier for this tenant.

## tftp

tftp - Configures and controls the TFTP server

### Synopsis

```
tftp disable <attribute>
tftp enable <attribute>

tftp get <attribute>
tftp set <attribute>=<value> [<attribute>=<value> ...]
tftp unset <attribute>
tftp show

tftp getStats

tftp serverLogs show
tftp serverLogs nlogs=<nlogs> logsize=<logsize>
```

### Description

The tftp command lets you configure the TFTP server in the cluster.

The serverLogs show command displays the number of log files and the maximum size for each file.

The serverLogs command allows you to set the two server logging parameters, nlogs and logsize. You can set one parameter or both. Changes occur only to the one or ones specified. When setting logsize, you can add the suffix K or M signify units of thousands or millions. Note that in order for these changes to take effect you must save the changes and restart the server Agent.

```
tftp serverLogs nlogs=6 logsize=500K
tftp serverLogs logsize=5M
```

The getStats command displays the requested TFTP server statistics.

### Examples

```
nrcmd> tftp enable docsis-access
nrcmd> tftp set file-cache-directory="CacheDir"
nrcmd> tftp reload
```

## Status

## See Also

[server](#)(nrcmd)

### Attributes

#### active-directory-domain [string](#)

Specifies the name of an Active Directory Domain the TFTP server will use to provide dynamic configuration file support.

#### default-device [string](#)

Specifies the name of the default disk device the TFTP server will use when none is specified in the pathname contained in the TFTP request. This property is specifically to be used on NT to specify a default drive letter.

#### file-cache [bool](#) default = disabled

Specifies whether the TFTP server should perform file caching on files located in the file-cache-directory.

#### file-cache-directory [string](#)

Specifies a path to a cache directory the TFTP server will use to find the files to put into cache. Upon start up the TFTP server will load all the files located in this directory into cache.

#### file-cache-max-memory-size [rangeint](#)(0-2147483647) default = 32000

Specifies the maximum number of bytes available to the server for file-caching.

#### home-directory [string](#)

Specifies a path to a home directory the TFTP server will use to resolve TFTP requests. If use-home-directory-as-root is disabled, the home directory is used in conjunction with the paths specified in the search-list to resolve TFTP requests.

#### initial-packet-timeout [time](#) default = 5s

Specifies the initial length of time, in seconds, the TFTP server will wait after sending a response to a client before declaring that response timed-out and sending a retransmission to the client. Note: This attribute is ignored if initial-packet-timeout-ms has been configured.

#### initial-packet-timeout-ms [int](#)

Specifies the initial length of time, in milliseconds, the TFTP server will wait after sending a response to a client before declaring that response timed-out and sending a retransmission to the client. Note: If this attribute is configured, the initial-packet-timeout attribute is ignored.

#### log-level [rangeint](#)(0-4) default = 3

Specifies the level of verbosity the TFTP server will employ when writing log messages to the TFTP server log file. Each integer value from 0 through 4 enables the following log levels: None, Error, Warning, Information and Activity.

#### log-settings [flags](#)(verbose=1, no-success-messages=2) default = verbose

Enables the TFTP server to allow control over additional details about the events listed in the log-settings. These details can help analyze a problem, but can also cause the log files to fill up quickly if left enabled for a long period of time. When log files fill up quickly, they also turn over frequently. So you might lose important information. By default, the TFTP server's logging is verbose. The possible flags are:

- 1 verbose  
Causes all the logging in the server to be active. A success message is printed to the log for each successful transfer.
- 2 no-success-messages  
Causes the single line message that is normally logged for every successful read from or write to the TFTP server to not appear. It affects logging only for successful file reads from the TFTP server.

#### max-inbound-file-size [string](#) default = 1024k

Specifies the maximum file size limit that the TFTP server will enforce for a file written to the TFTP server. Default units is in kilobytes. Use k, m or g to indicate kilobytes, megabytes or gigabytes.

#### min-socket-buffer-size [rangeint](#)(1-2147483647) default = 65536

Specifies the minimum socket buffer size the TFTP server will use for the well known port on which it is listening for TFTP requests.

**packet-trace-level** [rangeint](#)(0-4) default = 0

Specifies the level of verbosity the TFTP server will employ when writing messages to the server trace file. Each integer value from 1 through 4 enables increasing levels of tracing. Setting packet trace level to 0 disables tracing.

**port-number** [rangeint](#)(1-65535) default = 69

Specifies the UDP port number the TFTP server will use to listen for TFTP requests.

**read-access** [bool](#) default = enabled

Specifies how the TFTP server should respond to file read requests from TFTP clients. If this feature is disabled, the TFTP server will refuse file read requests.

**search-list** [string](#)

Specifies a comma separated list of paths the TFTP server will use to resolve TFTP requests. If use-home-directory-as-root is enabled, the paths in the search list are ignored and the home directory is used to resolve all TFTP requests.

**session-timeout** [time](#) default = 20s

Specifies the maximum length of time, in seconds, the TFTP server will wait after transmitting the initial response before giving up retrying on that response. If no response is received from the client within this timeout period, the TFTP session is terminated.

**use-home-directory-as-root** [bool](#) default = disabled

Specifies whether or not the TFTP server will treat pathnames contained within TFTP requests as if the paths were rooted at the specified home directory. If this feature is enabled, the TFTP server will attempt to resolve both absolute and relative pathnames to paths located beneath the specified home directory.

**write-access** [bool](#) default = disabled

Specifies how the TFTP server should respond to file write requests from TFTP clients. If this feature is disabled, the TFTP server will refuse file write requests.

**write-allow-file-create** [bool](#) default = disabled

Specifies whether to allow file creates on a PUT. If false and write-access equals true, the file must exist on the server for the PUT to succeed. If true and write-access is true, then the file does not need to exist on the server first and will be created on a PUT. max-inbound-file-size will apply a limit to the size of the file.

## tftp-interface

tftp-interface - Configures the network interfaces of the TFTP server

### Synopsis

```
tftp-interface <name> create [<attribute>=<value>]
tftp-interface <name> delete
tftp-interface list
tftp-interface listnames
tftp-interface listbrief
tftp-interface <name> show
tftp-interface <name> set <attribute>=<value> [<attribute>=<value> ...]
tftp-interface <name> get <attribute>
tftp-interface <name> unset <attribute>

tftp-interface <name> enable <attribute>
tftp-interface <name> disable <attribute>
```

### Description

The tftp-interface command configures network interfaces for use by the Network Registrar TFTP server. The TFTP interface logically represents the hardware interface (for example, an Ethernet or Token Ring network interface card) that the TFTP server uses. The TFTP server uses the configured address information to determine which interface to use to send and receive packets.

If there are no defined interfaces, the server discovers and uses all available interfaces on the system. When this list is present, the server only uses the available interfaces, if any, that match this list.

## Examples

## Status

## See Also

### Attributes

#### address [subnet](#)

The IP address and subnet mask of an interface that the TFTP server should use.

#### ip6address [prefix](#)

Specifies the IPv6 address and prefix length of one or more interfaces that the TFTP server should use.

#### name [string](#) required,unique

Specifies the user-assigned name of the TFTP server interface.

## trap-recipient

trap-recipient - Configures destinations for SNMP trap messages

## Synopsis

```
trap-recipient <name> create
trap-recipient <name> delete
trap-recipient <name> enable <attribute>
trap-recipient <name> disable <attribute>
trap-recipient <name> set <attribute>=<value> [<attribute>=<value> ...]
trap-recipient <name> unset <attribute>
trap-recipient <name> get <attribute>
trap-recipient <name> [show]
trap-recipient list
trap-recipient listnames
trap-recipient listbrief
```

## Description

The trap-recipient command configures management stations to which the Network Registrar SNMP server sends trap messages. The traps to be generated are set on the DNS and DHCP server traps-enabled attribute.

## Examples

```
nrcmd> trap-recipient example-recipient create
nrcmd> trap-recipient example-recipient set ip-addr=192.168.0.34
nrcmd> trap-recipient example-recipient set ip6address=2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e
```

## Status

## See Also

[dhcp](#), [dns](#), [addr-trap](#)



## Attributes

### agent-addr [ipaddr](#)

An IP address to use as the source agent-address in traps sent to this recipient.

### community [string](#)

The SNMP community string of this trap recipient

### ip-addr [ipaddr](#)

Specifies the IP address of this trap recipient.

### ip6address [ip6addr](#)

Specifies the IPv6 address of this trap recipient.

### port-number [int](#) default = 162

The optional IP port number of this trap recipient

### snmp-auth-password [clrtxt](#)

Specifies the password for account authentication.

### snmp-auth-secret [encdata](#)

Specifies the encrypted password for account authentication.

### snmp-engine-id [string](#)

Specifies the engine ID of the recipient, if required.

### snmp-priv-password [clrtxt](#)

Specifies the password for communication privacy.

### snmp-priv-secret [encdata](#)

Specifies the encrypted password for communication privacy.

### snmp-security [enumint](#)(no-auth=1, auth-nopriv=2, auth-priv=3) default = auth-priv

Specifies which security level to use.

no-auth  
No authentication or privacy.  
auth-nopriv  
Use SHA for account authentication. Requires an authentication password.  
auth-priv  
Use SHA for account authentication and AES for communication privacy. Requires both authentication and privacy passwords.

### snmp-trap-msg [enumint](#)(TRAP=1, INFORM=2) default = TRAP

Defines if this client wants a TRAP or INFORM message.

### snmp-trap-version [enumint](#)(SNMPv2c=2, SNMPv3=3) default = SNMPv2c

Specifies which SNMP protocol to use.

### snmp-user [string](#)

SNMP user name of this trap recipient.

### snmp-v3-protocol [enumint](#)(udp=0, tcp=1) default = udp

Specifies if this recipient should be sent messages over UDP or TCP.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

### v6-port-number [int](#) default = 162

The optional IPv6 port number of this trap recipient.

## update-policy

update-policy - Configures DNS update policies.

## Synopsis

```

update-policy <name> create
update-policy <name> delete
update-policy list
update-policy listnames
update-policy listbrief
update-policy <name> show
update-policy <name> get <attribute>
update-policy <name> unset <attribute>

update-policy <name> rules add "<value>" [<index>]
update-policy <name> rules remove <index>

update-policy < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
update-policy < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
update-policy <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The update-policy command lets you configure DNS update-policies. The most significant property of an update policy is an ordered list of rules. The rules are used to restrict or permit updates to DNS names. When adding a new rule, enclose the complete string in quotation marks. Use the backslash (\) to allow square brackets ([ ]) in the rule.

Note: If an update ACL has been configured on the zone, any update-policy configuration is ignored.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

```

nrcmd> update-policy test create
nrcmd> update-policy test rules add "grant any wildcard example* SRV"
nrcmd> update-policy test rules add "deny 1.1.1.1 wildcard \[a-z\]* A"

```

## Status

## See Also

### Attributes

#### description [string](#)

An optional description for this DNS update policy configuration.

#### name [string](#) required,unique

Specifies the name of the Update Policy.

#### rules [nlist\(obj\(0\)\)](#)

Lists rules that make up the update policy. Each rule has the following syntax:

```

action:    Can be grant or deny.
           grant - will allow an update if the rest of the rule
                matches.
           deny  - will deny an update if the rest of the rule
                matches.
acl-list:  A list of one or more ip addresses, network addresses,
           keys and/or named acl references. Note key names must be
           prefixed with "key " (i.e. "key key.example" ).
keyword:   Can be name, subdomain or wildcard.
           name - used to specify a specific RR.
           subdomain - used to specify a subdomain name.
           wildcard - used to specify a name with wildcard
                characters.
value:     The name, subdomain or wildcard value associated with the
           specified keyword. Note that all values specified are
           relative to the zone in which they are applied.
           Therefore it is not necessary to add the zone name to
           the end of the value.
           The supported wildcard characters are:
           * Will match zero or more characters. For example,
             the pattern dhcp-* matches all strings with the

```

```

    dhcp- prefix including the string dhcp-.
    ? Will match a single character. For example, the
      pattern zone?.com matches zonel.com, zone2.com,
      etc but does not match zone.com
    [...] Will match any characters listed within the
      brackets. For example, you can provide a range
      such as 0-9 or a-z. If the pattern also includes
      the - character, make it the first character in
      the list (i.e. dhcp[-a-z]*)
rr-types: A comma delimited list of RR types for this rule. Each RR
type can also be negated using the exclamation point
(i.e. !A,!TXT). You can also specify all types the an
asterisk (*).

```

tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

## vpn

vpn - Defines a logical VPN within which other DHCP objects may be configured

Note: The namespace command is a synonym for compatibility with earlier versions of Network Registrar.

### Synopsis

```

vpn <name> create <id-value> [<attribute>=<value> ...]
vpn <name> delete [-force]
vpn list
vpn listnames
vpn listbrief
vpn <name> show

vpn <name> get <attribute>
vpn <name> set <attribute>=<value> [<attribute>=<value> ...]
vpn <name> unset <attribute>

vpn < <name> | all > pull < ensure | replace | exact > <cluster-name>
    [-report-only | -report]
vpn < <name> | all > push < ensure | replace | exact > <cluster-list>
    [-report-only | -report]
vpn <name> reclaim <cluster-list> [-report-only | -report]

```

### Description

The vpn commands manipulate VPN objects in the DHCP server configuration. Each VPN must have a unique ID value. Other objects are associated with (or placed into) a VPN by VPN id.

Note: The reserved names "global" and "all" cannot be used as VPN names. These reserved names are used by other commands that take vpn names as arguments.

In contrast to most other CLI configuration objects, you can change the name of a VPN. The ID of the VPN, however, cannot be changed. Both the name and ID of the VPN must be unique upon creation.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

### Examples

```

nrcmd> vpn red show

nrcmd> vpn red set vpn-id=23:456

```

### Status

### See Also

## Attributes

### addr-blocks-default-selection-tags [nlist\(obj\(0\)\)](#)

Specifies the default selection tag (or list of tags) that will be associated with incoming subnet-allocation requests in this vpn that do not contain any subnet name data. No default.

### addr-blocks-use-client-affinity [bool](#)

Determines whether the DHCP server attempts to allocate subnets to clients using address-blocks that the clients have already used. Default is true (enable). If you disable this attribute, the server then supplies subnets from any block that is suitable, based on other selection data in the clients' messages.

### addr-blocks-use-lan-segments [bool](#)

Controls whether DHCP subnet-allocation uses the lan-segment attribute when configured on address-blocks.

### addr-blocks-use-selection-tags [bool](#)

Controls whether the server compares the subnet name data in incoming subnet-allocation requests with each address-block's selection tags. A block will only be considered if the two match. No default.

### description [string](#)

Describes the VPN that this object represents.

### id [int](#) required,unique,immutable

The VPN's unique id. This is a 32 bit value that is associated with every VPN-qualified IP address, subnet or address block. It is different than and unrelated to the standard 7-byte VPN-ID.

### name [string](#) required,unique

The VPN's name within the CNR management system. Independent from, but could be the same as the vrf-name.

### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

### tenant-private-network [bool](#) default = false

Indicates that this VPN represents the tenant non-routable (RFC1918) addresses on a local cluster. This attribute applies only to regional CCM clusters, and will be ignored if set on a local cluster.

### vpn-id [vpnid](#)

The vpn-id in RFC 2685 format (i.e., 7 octets), using a syntax similar to that used by IOS to enter the same information. The syntax is 3 hex octets, a colon, and 4 hex octets. For example 010203:04050607 would be the way to enter the following hex octets: 01:02:03:04:05:06:07 into this property.

### vrf-name [string](#)

The VPN's VRF name.

## zone

zone - configures a DNS zone

## Synopsis

```
zone <name> create primary file=<hostfile>
                        [protected | unprotected]
                        [<attribute>=<value>...]
zone <name> create primary <name server> <person>
                        [template=<template-name>]
                        [<attribute>=<value>...]
zone <name> create secondary <address> [<attribute>=<value>...]
zone <name> create from <src_view_name> [-omitRRs]
zone <name> delete
zone list [-primary | -secondary]
zone listnames [-primary | -secondary]
zone listbrief [-primary | -secondary]
zone <name> set <attribute>=<value> [<attribute>=<value> ...]
zone <name> get <attribute>
zone <name> unset <attribute>
```

```

zone <name> disable <attribute>
zone <name> enable <attribute>
zone <name> show

zone <name> addHost <host name> <address> [<alias> ...]
zone <name> removeHost <host name>
zone <name> showHost <host name>
zone <name> listHosts

zone <name> addRR [-sync] [-unprotected]
    <rr-name> [<tttl>] [IN] <type> <data> [<attribute>=<value> ...]
zone <name> addDNSRR <rr-name> [<tttl>] <type> <data>
    [<attribute>=<value> ...]
zone <name> removeRR <rr-name> [<type> [<data>]]
zone <name> removeDNSRR <rr-name> [<type> [<data>]]
zone <name> modifyRR <rr-name> <type> [<data>] <attribute>=<value>
    [<attribute>=<value> ...]
zone <name> findRR [-namePrefix <namePrefix>]
    [-rrTypes <rrTypeList>] [-protected | -unprotected]
zone <name> listRR [-protected | -unprotected | -all]

zone <name> rrSet <rr-name> [set host-health-check=off|ping|gtp-echo]
    [get host-health-check] [unset host-health-check] [show]

zone <name> < protect-name|unprotect-name > <rr-name>
zone <name> forceXfer secondary
zone <secondary-zone-name> promote-to-primary

zone <name> scavenge

zone <name> getScavengeStartTime

zone <name> applyTemplate <template-name>

zone <name> getstatus

zone <name> getUtilization

zone <name> ha-sync-all-rrs

zone <name> sync < update | complete > [-report-only | -report]

```

## Description

The zone command lets you create and edit DNS zones.

The name of the zone may be an IPv4 subnet (<address>/<length>), IPv6 prefix (<address>/<length>), prefix name (the prefix address is used), or view qualified DNS name (<view\_name>/<DNS name> or simple DNS name.

```

zone <name> create from <src_view_name> [-omitRRs]
    This command copies a zone from one DNS View to another, including
    all its associated RRs unless the -omitRRs option is specified.

```

```

zone <name> addHost <host name> <address> [<alias> ...]
zone <name> removeHost <host name>
zone <name> showHost <host name>
zone <name> listHosts

```

The addHost command adds a host with a given name, address and optional aliases to the zone.

The removeHost command removes a host from the zone.

The showHost command shows the specified host in the zone.

The listHosts command lists the hosts in the zone.

```

zone <name> addRR [-sync] [-unprotected]
    <rr-name> [<tttl>] [IN] <type> <data> [<attribute>=<value> ...]
zone <name> addDNSRR
    <rr-name> [<tttl>] <type> <data> [<attribute>=<value> ...]
zone <name> removeRR <rr-name> [<type> [<data>]]
zone <name> removeDNSRR <rr-name> [<type> [<data>]]
zone <name> modifyRR <rr-name> <type> [<data>]
    <attribute>=<value> [<attribute>=<value> ...]
zone <name> findRR [-namePrefix <namePrefix>]
    [-rrTypes <rrTypeList>] [-protected | -unprotected]
zone <name> listRR [-protected | -unprotected | -all]

```

The addRR command adds a resource record to a zone.

If the resource record defines a new nameset, it will be created as a protected name, unless the -unprotected flag is set.

If the new resource record is added to an existing nameset,

the protection state will remain unchanged.

Optional attributes such as order and weight can also be set on the new resource record. To set extra attributes, the data section must be enclosed in quotes when it contains an '=' character. Embedded quotes must also be escaped (i.e.: \") if this is the case. When connected to server versions prior to 8.2, attempts to add a protected record to an unprotected name will fail.

The removeRR command removes all specified resource records. Resource records may be specified by name, by name and type, or by name, type, and data (the data is specified in BIND-style format.) When connected to server versions prior to 8.2, the removeRR command can only be used to remove protected records.

The modifyRR command sets the specified attributes for the matching resource record. For example:

```
nrcmd> zone example.com. modifyRR a1 A 10.10.1.1 rdata=10.10.10.10 ttl=3600
```

Resource records are specified by name and type, or by name, type, and data if there are multiple entries for the given type. The data section must be enclosed in quotes when it contains an '=' character and embedded quotes must also be escaped (i.e.: \") if this is the case.

The addDNSRR command creates an unprotected resource record. The name, type, and data must be specified. If the new resource record is added to an existing protected nameset, the protection state will be ignored and the nameset will remain protected. Optional attributes such as order and weight can also be set on the new resource record. To set extra attributes, the data section must be enclosed in quotes when it contains an '=' character. When connected to server versions prior to 8.2, attempts to add a unprotected record to a protected name will fail.

The removeDNSRR command removes all specified resource records. Resource records may be specified by name, by name+type, or name+type+data. The changes take effect immediately; no server reload is necessary. When connected to server versions prior to 8.2, the removeDNSRR command can only be used to remove unprotected records. Also, if the DNS server is not running, the command will fail.

The findRR command displays the resource records matching a name prefix, a list of resource record types, and whether protected or not (or either).

The listRR command lists the resource records in the zone. RRs can be filtered using option -protected/-unprotected/-all. Default behaviour is to list both protected and unprotected RRs of the zone.

The arguments to commands that accept full or partial resource records use the same format as BIND files, with one difference for rr-names. BIND specifies that rr-names that do not end in a dot have the zone name appended. NRCMD will not append the zone name when the rr-name already ends in the zone name because that would produce an rr-name with the zone name repeated twice. In cases where the rr-name should repeat the zone name twice at the end, you must specify this explicitly, though the terminal dot can be omitted. It is recommended to use fully qualified rr-names that include the terminal dot to avoid any confusion.

**zone <name> protect-name|unprotect-name> <rr-name>**

The protect-name/unprotect-name command sets the protection status of the resource records for the name. Protected names cannot be updated using DNS update requests.

**zone <name> rrSet <rr-name> [set host-health-check=off|ping|gtp-echo] [get host-health-check] [unset host-health-check] [show]**

The rrSet command sets/unsets the host-health-check flag on resource records for the rr-name. When this flag is set, the A and AAAA record's health will be monitored.

When host-health-check is set to ping, ICMP protocol will be used to find host health.

When host-health-check is set to gtp-echo, GTP-C v2 protocol will be used to find host health.

**zone <name> forceXfer secondary**

The forceXfer command forces a full zone transfer of a secondary zone, regardless of the zone serial number, to

synchronize DNS data store. If a normal zone transfer is already in progress, the forceXfer command is scheduled immediately after the normal zone transfer finishes. An option for primary zones has not been implemented yet.

**zone <name> scavenge**

**zone <name> getScavengeStartTime**

The scavenge command schedules zone scavenging immediately for the given zone regardless of the scavenging interval.

The getScavengeStartTime command returns the date and time of the next check for stale records, when records might be scavenged.

**zone <secondary-zone-name> promote-to-primary**

The promote-to-primary command can be used to promote a secondary zone to a primary zone (for example, if the primary DNS server has had a hardware failure).

**zone <name> applyTemplate <template-name>**

The applyTemplate command applies the specified zone template to the zone. All properties except view-id (based on below condition) configured on the zone template are applied to the zone.

Condition: If zone is in 'Default' DNS Views then only view-id set on template would be used.

**zone <name> getstatus**

The getstatus command can be used to get the status of the specified zone. This command is valid only for primary and reverse zones.

**zone <name> getUtilization**

The getUtilization command can be used to get the count of total number of A and AAAA record for the specified zones.

**zone <name> ha-sync-all-rrs**

The ha-sync-all-rrs command can be used to manually schedule HA zone synchronization for the zone, or to raise its priority, if the zone is already in the sync-pending state. This command can only be used on the HA main server, when the server is operating in the normal HA state.

**zone <name> sync < update | complete >**

Only available when connected to a regional cluster.

To export a zone, see the **export zone** command.

To export a list of the zone names, see the **export zonenames** command.

## Examples

```
nrcmd> zone example.com. create primary file=host.local
nrcmd> zone example.com. create primary ns ns-server
```

## Status

## See Also

[export dnssec-ds](#), [export zone](#), [export zonenames](#), [zone-template](#)

## Attributes

**alternate-view-ids** [nlist\(obj\(0\)\)](#)

Specifies the alternate view identifiers for this zone.

**defttl** [rangetime\(0-68y5w3h14m7s\)](#) default = 24h

Controls the default TTL value used for resource records in this zone that do not specify a TTL.

**description** [string](#)

An optional description for this zone.

**dist-map** [objref\(0\)](#)

Identifies the zone distribution map associated with the specified zone. The zone distribution map describes which primary and secondary DNS servers should provide DNS service for this zone.

**dnssec** [bool](#) default = false

Indicates DNSSEC processing is enabled for this zone. The attribute must be enabled only after the zone has been published. This feature cannot be enabled with RPZ.

**ds-rr** [string](#)

Specifies the DS RR for this zone using the current Key Signing Key. This record must be published in the parent zone for this zone.

**expire** [rangetime](#)(1s-68y5w3h14m7s) default = 1w

Sets the number of seconds that a secondary server can continue providing zone data without confirming that the data remains current. The expire interval must be greater than the refresh interval.

**key-group** [tag](#)

Specifies the name of the DNSSEC key-group for this zone.

**minttl** [rangetime](#)(0-68y5w3h14m7s) default = 10m

Specifies the TTL value that Caching DNS servers should use for caching negative responses.

**nameservers** [nlist](#)(obj(0)) required

Lists the nameservers for this zone.

**notify** [enumby](#)t default = notify-all

Enables notification to other authoritative servers when this zone changes.  
Possible settings are:  
disabled  
Notifications will be turned off.  
notify-all  
Notifications will be sent to all NS and notify-list servers.  
notify-ns  
Notifications will only be sent to NS servers.  
notify-list  
Notifications will only be sent to the notify-list servers.  
Note that if set, the DNS server global 'notify' value will override the default value.

**notify-list** [nlist](#)(obj(0))

Lists IPv4 and IPv6 addresses to notify of changes to this zone. An optional port can be specified for any given address using the format @.  
For example,

192.168.1.2@50053, 2001:db8:85a3:0:0:8a2e:370:7334

Note that notify attribute must be set to 'notify-all' or 'notify-list' in order for this setting to take effect.

**ns** [dname](#) required

Displays the fully-qualified domain name of the primary name server for this zone. This host is the original, or primary source, of data for this zone.

**nsttl** [dnsttl](#)

Displays the ttl value applied to the NS resource records of the zone.

**origin** [dname](#) required,immutable

Displays the fully-qualified name of the zone root, also known as the domain name.

**owner** [objref](#)(0)

Names the owner of this zone. Use the owner field to group similarly owned zones and to limit administrative access.

**person** [dname](#) required

Displays a domain name specifying the mailbox of the person responsible for this zone. The first label is a user or mail alias, the rest of the labels are a mail destination. A mailbox of hostadmin@example.com would be represented as hostadmin.example.com.

**refresh** [rangetime](#)(1s-68y5w3h14m7s) default = 3h

Sets the number of seconds that a secondary server waits before polling for zone changes and refreshing its zone data.



**region** [objref](#)(0)

Associates a region with the specified object. Use the region field to group similarly located zones and to limit administrative access.

**restrict-query-acl** [amelist](#) default = any

Specifies the zone access control list (ACL) used to restrict the queries that the DNS server for this zone accepts. This list can contain host IPs, network addresses, TSIG keys, and (global) ACLs. Only queries from clients defined in the ACL are accepted. Note that if set, the DNS server global 'restrict-query-acl' value will override the default value.

**restrict-xfer** [bool](#) default = true

Restricts sending zone transfers to a specific set of hosts. If you restrict zone transfers, use the restrict-xfer-acl attribute to specify the access control list that defines which servers can perform zone transfers. Note that if set, the DNS server global 'restrict-xfer' value will override the default value.

**restrict-xfer-acl** [amelist](#) default = none

Specifies the access control list designating who can receive zone transfers from this zone. Note that if set, the DNS server global 'restrict-xfer-acl' value will override the default value.

**retry** [rangeint](#)(1s-68y5w3h14m7s) default = 60m

Sets the number of seconds that a secondary server waits before it retries polling for changes to zone data or it retries a zone transfer that has encountered errors. The retry interval must be less than (expire - refresh).

**round-robin** [bool](#) default = true

Specifies whether you want round-robin cycling of equivalent records in responses to queries. Equivalent records are records of the same name and type. Since clients often only look at the first record of a set, enabling this feature can help balance load. Note that if set, the DNS server global 'round-robin' value will override the default value.

**rpz** [bool](#) default = false

Specifies if the zone RR list contains DNS Firewall RPZ rule data. This feature cannot be enabled with DNSSEC.

**scvg-enabled** [bool](#) default = false

Enables dynamic resource-record scavenging for the zone. This attribute removes stale records when clients are configured to perform DNS updates but do not delete their entries when they're no longer valid. If the DHCP server is used to perform updates, it will also delete records when client leases expire. If enabling scavenging on DHCP updated zones, ensure that the scvg-refresh-interval is set to be longer than the lease expiration time.

**scvg-interval** [rangeint](#)(60m-1y)

Sets the interval at which the scavenging process will run for the zone. If unset, the DNS server setting scvg-interval will be used.

**scvg-max-records** [rangeint](#)(1-10000)

Sets the maximum number of records that can be removed from a zone during its scavenging interval. If unset, the DNS server scvg-max-records attribute will be used.

**scvg-refresh-interval** [rangeint](#)(60m-1y)

When scavenging is enabled, specifies the expiration time for RRs. The expiration time is calculated based on the time the RR was last modified. Once an RR has expired, it becomes a candidate for scavenging and may be deleted by the scavenging process. If unset, the DNS server scvg-refresh-interval will be used. Updates to the RR (i.e. dynamic updates, prereq only updates, etc.) will reset the record's modification time and thus recalculate the scavenging expiration time. NOTE: RRs in a protected name set are not scavenged.

**serial** [rangeint](#)(1--1) required

Displays an administratively specified serial number. The serial number value must always increase; therefore, this serial number is only applied to the zone if it is greater than the actual (dynamic) serial number.

**soattl** [dnsttl](#)

Displays the time-to-live (ttl) value applied to the SOA resource record of the zone.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**update-acl** [amelist](#) default = none

Specifies the access control list for DNS updates to the zone, given as an address match element list. The access control list is not applied to administrative edits managed through the CCM server. Note that if set, the DNS server global 'update-acl' value will override the default value.

**update-policy-list** [nlist](#)(obj(0))

Lists the DNS update policies used to authorize or deny DNS updates. This attribute is ignored if the update-acl attribute is also set.

**view-id** [int](#) default = 0

Specifies the view identifier for this zone.

## zone-dist

zone-dist - Configures zone distributions

### Synopsis

```
zone-dist <name> create <primary-cluster> [<attribute>=<value> ...]
zone-dist <name> delete
zone-dist list
zone-dist listnames
zone-dist listbrief
zone-dist <name> show

zone-dist <name> get <attribute>
zone-dist <name> set <attribute>=<value> [<attribute>=<value> ...]
zone-dist <name> unset <attribute>

zone-dist <name> addSecondary <secondary-cluster>
    [<primary-server-ipkeys>]
zone-dist <name> removeSecondary <secondary-cluster>
    [<primary-server-ipkeys>]
zone-dist <name> listSecondaries

zone-dist <name> addCachingServer <caching-cluster>
    [<primary-servers>]
zone-dist <name> removeCachingServer <caching-cluster>
zone-dist <name> listCachingServers

zone-dist <name> sync < update | complete | exact >
    [no-rrs] [primary-only] [-report-only | -report]
```

### Description

The zone-dist command lets you define and manage zone distribution configurations.

On local clusters, the zone-dist sync command synchronizes staged edits to the DNS server and synchronizes primary zones to secondaries. Regardless of the mode selected, the exact list of authoritative zones (primary and secondary) is synchronized with the DNS server.

On the regional cluster, the zone-dist sync command synchronizes primary zones from the regional configuration to the primary local cluster, and synchronizes primary zones to secondaries. Primary zones on the local cluster are replaced in Update or Complete mode. In Exact mode, extra primary zones found on the local cluster are deleted.

Secondary servers use the same synchronization logic at both local and regional clusters. In Update mode, synchronization ensures only that corresponding secondary zones exist on the server. In Complete mode, any existing zones are updated to use the primary servers list specified by the distribution map. In Exact mode, any zones not matching the distribution map are deleted.

Use the [no-rrs] and [primary-only] flags to skip portions of the synchronization logic. While using these flags might improve the performance of the command, use them only when you are

certain there are no changes pending. For example, if primary zones are current with the DNS server, you can use the [no-rrs] flag to synchronize your secondary zones.

On local clusters, RRs are always synchronized to the local DNS server if changes are pending, and thus the [no-rrs] flag is ignored.

Caching server commands help in configuring CDNS server. It provides propagation of view list and creation of exception to the CDNS server.

## Examples

## Status

## See Also

[cluster](#)

## Attributes

### description [string](#)

An optional description for this Zone Distribution configuration.

### name [string](#) default = Default, required,unique

Names this zone distribution map.

### notify [enumint](#)(enable=1, disable=2, use-server-settings=3, use-primary-zone-settings=4, enable-ns=5, enable-notify-list=6) default = use-server-settings

Specifies how the 'notify' property should be set on the secondary zones.

- 1 enable  
Sets the value to notify-all, which sends notifications to the NS and notify-list servers when the zone changes.
- 2 disable  
Sets the value to disabled, which prohibits notification to other authoritative servers when the zone changes.
- 3 use-server-settings  
Unsets the value, which allows the secondary DNS server global 'notify' value to apply.
- 4 use-primary-zone-settings  
Sets the value from the primary zone.
- 5 enable-ns  
Sets the value to notify-ns, which only sends notifications to the NS servers when the zone changes.
- 6 enable-notify-list  
Sets the value to notify-list, which only sends notifications to the notify-list servers when the zone changes.

### notify-list [nlist](#)(obj(0))

Lists IPv4 and IPv6 addresses to notify of changes to this zone. An optional port can be specified for any given address using the format @.  
For example,

```
192.168.1.2@50053, 2001:db8:85a3:0:0:8a2e:370:7334
```

Note that notify attribute must be set to 'notify-all' or 'notify-list' in order for this setting to take effect.

### primary [oid](#)(0)

Identifies the cluster or HA DNS pair serving the primary zones associated with this zone distribution map.

### primary-servers [nlist](#)(obj(0))

Lists the primary-servers to set when creating a secondary zone or caching server exception. Each server IPv4 or IPv6 address can be followed with an optional port and/or key name. Specifying the key name allows the server to be configured for secure zone transfers. Validation: A list of one or more ip addresses with an optional port and/or key name (format [@[[-]]).

For example:  
192.168.1.2@50053, 2001:db8:85a3:0:0:8a2e:370:7334-key.example.com.

### restrict-query [enumint](#)(use-map-settings=1, use-server-settings=2, use-primary-zone-settings=3) default = use-primary-zone-settings

Specifies how the 'restrict-query-acl' property should be set on the secondary zones.

- 1 use-map-settings  
Uses the 'restrict-query-acl' value in this object to configure the secondary zones.

- 2 use-server-settings  
Unsets the value, which allows the secondary DNS server global 'restrict-query-acl' value to apply.
- 3 use-primary-zone-settings  
Sets the value from the primary zone.

#### restrict-query-acl [amelist](#)

Specifies the secondary zone access control list (ACL) used to restrict the queries that the DNS server will accept, if 'restrict-query' is set to use-map-settings. This list can contain host IPs, network addresses, TSIG keys, and (global) ACLs. Only queries from clients defined in the ACL are accepted.

#### restrict-xfer [enumint](#)(enable=1, disable=2, use-server-settings=3, use-primary-zone-settings=4) default = use-primary-zone-settings

Specifies how the 'restrict-xfer' property should be set on the secondary zones.

- 1 enable  
Sets the value to enabled, which limits zone transfers to a specific set of hosts. You must then use the 'restrict-xfer-acl' attribute to list the servers allowed to perform zone transfers.
- 2 disable  
Sets the value to disabled, which places no restrictions on zone transfers.
- 3 use-server-settings  
Unsets the value, which allows the secondary DNS server global settings to apply.
- 4 use-primary-zone-settings  
Sets the value from the primary zone.

#### restrict-xfer-acl [amelist](#)

Identifies the access control list designating who can receive zone transfers on the secondary zone, when restrict-xfer is enabled.

#### round-robin [enumint](#)(enable=1, disable=2, use-server-settings=3, use-primary-zone-settings=4) default = use-primary-zone-settings

Specifies how the 'round-robin' property should be set on the secondary zones.

- 1 enable  
Enables round-robin for the secondary zone.
- 2 disable  
Disables round-robin for the secondary zone.
- 3 use-server-settings  
Uses the server setting. Unset the value set on secondary zones.
- 4 use-primary-zone-settings  
Uses the value set at the primary zone to configure the secondary zone.

#### tenant-id [short](#) default = 0, immutable

Identifies the tenant owner of this object.

#### use-gss-tsig [bool](#) default = false

Specifies the secondary zone setting when gss-tsig should be used to request zone transfers from the primary server.

## zone-template

zone-template - Configures a zone template

### Synopsis

```

zone-template list
zone-template listnames
zone-template listbrief

zone-template <name> create [<attribute>=<value> ...]
zone-template <name> delete
zone-template <name> set <attribute>=<value> [<attribute>=<value> ...]
zone-template <name> get <attribute>
zone-template <name> unset <attribute>

zone-template <name> disable <attribute>
zone-template <name> enable <attribute>
zone-template <name> show

zone-template <name> create clone=<clone-name>
zone-template <name> apply-to [all | <[view1/]zone1>[,...]]

zone-template < <name> | all > pull < ensure | replace | exact >
    <cluster-name> [-report-only | -report]
zone-template < <name> | all > push < ensure | replace | exact >
    <cluster-list> [-report-only | -report]
zone-template <name> reclaim <cluster-list> [-report-only | -report]

```

## Description

The zone-template command lets you configure templates to use when creating zones.

The pull, push, and reclaim commands are only available when connected to a regional cluster. For push and reclaim, a list of clusters or "all" may be specified.

## Examples

## Status

## See Also

[zone](#)

## Attributes

**alternate-view-ids** [nlist](#)(obj(0))

Specifies the alternate view identifiers for this zone.

**defttl** [rangetime](#)(0-68y5w3h14m7s)

Controls the default TTL value used for resource records in a zone that do not specify a TTL.

**description** [string](#)

An optional description for this Zone template configuration.

**dist-map** [oid](#)

Associates a zone distribution map with a zone. The map describes the primary and secondary DNS servers that provide DNS service for this zone.

**dnssec** [bool](#) default = false

Enables DNSSEC processing for this zone.

**expire** [rangetime](#)(1s-68y5w3h14m7s)

Sets the number of seconds that a secondary server can continue providing zone data without confirming that the data remains current. The expire interval must be greater than the refresh interval.

**key-group** [tag](#)

Specifies the name of the DNSSEC key-group for this zone.

**minttl** [rangetime](#)(0-68y5w3h14m7s)

Specifies the TTL value that Caching DNS servers should use for caching negative responses.

**name** [string](#) required,unique

Names this zone template.

**nameservers** [nlist](#)(obj(0))

Lists the nameservers for a zone.

**notify** [enumbyt](#)

Enables notification to other authoritative servers when this zone changes.  
Possible settings are:  
disabled  
Notifications will be turned off.  
notify-all  
Notifications will be sent to all NS and notify-list servers.  
notify-ns  
Notifications will only be sent to NS servers.  
notify-list  
Notifications will only be sent to the notify-list servers.

**notify-list** [nlist](#)(obj(0))

Lists IPv4 and IPv6 addresses to notify of changes to this zone. An optional port can be specified for any given address using the format @. For example,

192.168.1.2@50053, 2001:db8:85a3:0:0:8a2e:370:7334

Note that notify attribute must be set to 'notify-all' or 'notify-list' in order for this setting to take effect.

#### **ns** [string](#)

Specifies the fully-qualified domain name (FQDN) of the primary name server for the zone. This host is the original or primary source of data for this zone. Note: Network Registrar treats the value set here as a string rather than a dnsname to enable encoding relative or absolute names in this attribute.

#### **nsttl** [dnsttl](#)

Controls the ttl value applied to the zone NS resource records.

#### **owner** [objref\(0\)](#)

Identifies the owner of this zone. Use the owner field to group similarly owned zones and to limit administrative access.

#### **person** [string](#)

Specifies the mailbox for the hostadmin (person) in domain name form. The first label is a user or mail alias, the rest of the labels are a mail destination. A mailbox of hostadmin@example.com would be represented as:  
hostadmin.example.com.

Note: Network Registrar treats the value set here as a string rather than a dnsname to enable encoding relative or absolute names in this attribute.

#### **refresh** [rangetime\(1s-68y5w3h14m7s\)](#)

Sets the interval at which a secondary server contacts its primary server for changes to zone data. The interval is defined in the server SOA record and is also known as the secondary refresh time.

#### **region** [objref\(0\)](#)

The region associated with this object. This region field is used to group similarly located zones and can be used to limit administrative access.

#### **restrict-query-acl** [amelist](#)

Specifies the zone access control list (ACL) used to restrict the queries that the DNS server for this zone accepts. This list can contain host IPs, network addresses, TSIG keys, and (global) ACLs. Only queries from clients defined in the ACL are accepted.

#### **restrict-xfer** [bool](#)

Restricts sending zone transfers to a specific set of hosts. If you restrict zone transfers, you need to use the restrict-xfer-acl property to specify the access control list of who is allowed to perform zone transfers.

#### **restrict-xfer-acl** [amelist](#)

Identifies the access control list designating who can receive zone transfers from this zone.

#### **retry** [rangetime\(1s-68y5w3h14m7s\)](#)

Sets the amount of time that a secondary server waits before it retries polling for changes to zone data or it retries a zone transfer that has encountered errors. The retry interval must be less than the expire and refresh intervals. A good value is between one-third and one-tenth of the refresh time.

#### **round-robin** [bool](#)

Specifies whether you want round-robin cycling of equivalent records in responses to queries. Equivalent records are records of the same name and type. Since clients often only look at the first record of a set, enabling this features can help balance load.

#### **scvg-enabled** [bool](#)

Enables dynamic resource-record scavenging for the zone. This attribute removes stale records when clients are configured to perform DNS updates but do not delete their entries when they're no longer valid. If the DHCP server is used to perform updates, it will also delete records when client leases expire. If enabling scavenging on DHCP updated zones, ensure that the scvg-refresh-interval is set to be longer than the lease expiration time.

#### **scvg-interval** [rangetime\(60m-1y\)](#)

Sets the interval at which the scavenging process will run for the zone. If unset, the DNS server setting `scvlg-interval` will be used.

**scvlg-max-records** [rangeint](#)(1-10000)

Sets the maximum number of records that can be removed from a zone during its scavenging interval. If unset, the DNS server `scvlg-max-records` attribute will be used.

**scvlg-refresh-interval** [rangetime](#)(60m-1y)

When scavenging is enabled, specifies the expiration time for RRs. The expiration time is calculated based on the time the RR was last modified. Once an RR has expired, it becomes a candidate for scavenging and may be deleted by the scavenging process. If unset, the DNS server `scvlg-refresh-interval` will be used. Updates to the RR (i.e. dynamic updates, prereq only updates, etc.) will reset the record's modification time and thus recalculate the scavenging expiration time.  
NOTE: RRs in a protected name set are not scavenged.

**serial** [rangeint](#)(1--1)

Sets the starting serial number of the zone. A DNS server uses a serial number to indicate database changes. Increments to this number trigger zone transfers to a secondary server.

**soattl** [dnsttl](#)

Controls the ttl value applied to a zone SOA resource record.

**tenant-id** [short](#) default = 0, immutable

Identifies the tenant owner of this object.

**update-acl** [amelist](#)

Specifies the access control list for DNS updates to a zone, defined as an address match element list. The access control list is not applied to dynamic updates coming from the UIs. Updates from them UIs are always allowed as long as the zone attribute `dynamic` is enabled. The access control list is not applied to administrative edits managed through the CCM server.

**update-policy-list** [nlist](#)(obj(0))

An ordered list of DNS update policies that can be used to authorize or deny DNS updates. This attribute will be ignored if `update-acl` is also set.

**view-id** [int](#)

Specifies the view identifier for this zone.

## Using the *nrcmd* Program in Scripts

You can use the *nrcmd* command to interactively configure and control a Cisco Network Registrar cluster, or you can use it as a programming interface for another program or script.

### Connecting to Network Registrar

When you use the *nrcmd* command, you connect to a Network Registrar *cluster* to read and write configuration data, read state data, and perform control operations.

A Network Registrar cluster consists of:

- The data manager, the CCM server, which controls access to persistent datastores that contain configuration and state information for the DNS, DHCP, and TFTP servers.
- The server agent, AIC Server Agent, which starts and stops the protocol servers, and provides a standard control interface to them.
- The DNS, DHCP, and TFTP protocol servers.

### Performing Authentication

When you log in to a cluster you are authenticated with a name and a password. The authentication protocol uses one-way hashes so that a password does not travel across the wire. In interactive mode, the *nrcmd* command prompts for a valid username and password. You can also provide the username or password on the command line or using the environment variables.

Because you may not want to embed the administrator password in a command script, the environment variables provide alternate locations with different visibility levels. The environment variables CNR\_CLUSTER, CNR\_NAME, and CNR\_PASSWORD specify the cluster, administrator name and administrator password values, respectively.

### Choosing Scripting Techniques

Because *nrcmd* does a significant amount of processing at connect time, it is more efficient to perform multiple commands in a single session rather than to initiate a distinct connection and login for each command. The simplest way to have a single *nrcmd* session perform multiple commands is to create a batch file with one command per line and to redirect standard input from that file. A more complicated approach, but one that provides more control over the command sequence, is to run *nrcmd* from a controlling program and have that program send commands and read their status and output.

### Using *nrcmd* Batch Files

The simplest way to automatically perform multiple configuration commands is to create a batch file of *nrcmd* commands and have them executed sequentially. For example, to create a scope and add reservations to it, you can enter these commands and store them in the file `scope.txt`. Lines beginning with the pound character (#) are comment lines:

```
# This set of commands creates a scope and adds four reservations
scope demo1 create 24.10.2.0 255.255.255.0
scope demo1 addReservation 24.10.2.1 1,6,0a:23:45:67:89:01
scope demo1 addReservation 24.10.2.2 1,6,0c:23:45:67:89:02
scope demo1 addReservation 24.10.2.3 1,6,0c:23:45:67:89:03
scope demo1 addReservation 24.10.2.4 1,6,0a:23:45:67:89:04
```

Note: End the last command line with a newline character, or the command will not be executed.



You can then run a single *nrcmd* session to execute all of these commands.

```
% nrcmd -b < scope.txt
```

The advantage to using batch files is that you can execute multiple configuration commands while only incurring the connection cost once. However, if a command fails (such as the initial scope creation in the previous example), the batch file continues even though subsequent commands are now useless.

### Command Syntax

When you execute *nrcmd* commands that contain equal-signs, you must put them within quotation marks. For example, to use a single command to create a client-class name, enter:

```
nrcmd -C cluster -N name -P password "client MAC create client-class-name= name "
```

### Adding Program Control

A more sophisticated method for automatically configuring and controlling Network Registrar is to have a program or script start a *nrcmd* session and communicate with the session through standard input and output.

To control *nrcmd* from another program, you need to start *nrcmd* from the controlling program and redirect standard input and output from *nrcmd* to file handles in the controlling program. The controlling program can then write commands to the input file handle and read results from the output file handle.

When running in batch mode, *nrcmd* reads a line of input at a time and prints a new line after the prompt. This provides an easily parsed sequence of lines in response to any command where:

- The syntax is *status-line result-lines prompt-line*
- The *status-line* has the format `[0-9]{3}.*`.
- There may be zero or more *result-lines* of any format.
- The *prompt-line* is `nrcmd> .`

The exact details of starting up *nrcmd* as a child process, and writing to and reading from its standard input and output, are specific to the programming language you use to implement the controlling program.

## CLI Codes and Formats

[Status Returns](#)

[Network Registrar Error Codes](#)

[Import and Export File Formats](#)

### Status Returns

The **nrcmd** program returns status information on the first line of information written to the standard output stream. If there is more data, **nrcmd** displays this information on additional lines.

The first line consists of a numeric status that is followed by a human-readable error status.

The status codes are all three-digit integer decimal numbers. The range 100 through 599 is grouped as in Table 4-1.

**Table 4-1. Status Codes**

Value	Description
100-199	Normal return
200-299	Informational (warning)
300-499	Error
500-599	Fatal Error

For anything other than an error, Network Registrar assumes that the requested operation was completed; however, some warning messages signal a condition that must be corrected. Unless a fatal error occurs, the command line interface will keep running in interactive mode. Fatal errors imply that something serious happened and that you must restart the Network Registrar command line processor.

### Network Registrar Error Codes

Table 4-2 lists and describes the Network Registrar error codes.

**Table 4-2. Error Codes**

Number	Description
100	OK

101	OK, with warnings
102	Lease already reserved to this client
103	Lease not reserved
104	Lease deleted along with reservation
105	Lease created along with reservation
108	Ok, with duplicate objects updated
109	Ok - resource status is ...
204	Assigned host is not contained in pool zone
300	Unexpected error
301	No server found
302	Not found
303	Read only property
304	No policy found
305	Too many
306	Unknown command
307	Unknown keyword
308	Unknown parameter
309	Too many arguments

310	Too few arguments
311	No response to lease request
312	Unexpected response from server
313	No match
314	Duplicate object
315	Import failure
316	Invalid
317	Open failed
318	No MAC address found
319	No lease found
320	Generic error
321	Invalid name
322	Feature not supported
323	Read error
324	Invalid IP address list
325	Invalid type
326	ODBC database access error
327	IP address not contained within pool subnet

328	Identical MX resource record already exists
329	Identical TXT resource record already exists
330	Address is not contained in pool
331	Host is already assigned to an address
332	Address is already assigned to a host
333	No unassigned IP address found in pool
334	Address has not been assigned to a host
335	Static address pools are not enabled, create a pool to enable
336	Range overlaps another pool
337	Host has multiple IP address assignments
338	Address has multiple host assignments, must supply <name> argument
339	Expected unsigned 16-bit preference value
340	ODBC 3.x or higher required
350	Generic DHCP error
351	Cannot resolve partner name to an IP address
352	No failover object for specified partner
353	Still communicating with partner
354	Server is already in partner-down state

355	Can't set partner-down while in recover
356	Not allowed in read-only mode
357	Not a secondary
358	Not a primary
359	No zone matched
360	Forcexfer for this zone is already scheduled
361	Lease is not reserved
362	Scope unknown in server
363	Invalid IP address in server
364	Invalid MAC address in server
365	Failure creating MAC address in server
366	Unknown object in server
367	Command not supported by server
368	Bad length in server
369	Inconsistent scope in server
370	updateSMS not configured in server
371	Server out of memory
372	SMS interface .dll did not load correctly

373	updateSMS already processing in server
374	Invalid word array
375	updateSMS invalid argument
376	Lease is reserved to a different client
377	Client already reserved a different lease
378	Field name or number not found
379	Suboption name or number already exists
380	Suboption name or number not found
381	Invalid character '-' in vendor-option name
382	Data not found for vendor-option
383	Field name or number already exists
384	counted-array can only be used with array types
385	Read-only attribute cannot be modified
386	Required attribute cannot be unset
387	Invalid vpn-id
388	Invalid IP address syntax
389	Invalid vpn name
390	Invalid IP address value

391	Invalid vpn specification
392	Cannot remove session's current vpn
393	Duplicate property
394	Invalid vpn-id specification
395	Lease has no scope pointer -- internal error
396	Invalid vpn-id specification
397	Expression too long
398	Insufficient memory
399	Invalid DNS Views
401	Login failure
402	Permission denied
403	Couldn't lock database
404	Login required
405	Invalid license key
406	A lock is required for this operation
407	Unable to release lock
408	Unable to obtain lock
409	Couldn't lock static address pool



501	Connection failure
502	Server failure
503	Cluster version failure
504	Wrong license key type, local cluster key type needed
504	Wrong license key type, regional key type needed
505	Connected to wrong cluster type
506	Duplicate option id
507	Duplicate option name
508	Duplicate vendor-option-enterprise-id
509	Duplicate vendor-option-string
510	Version mismatch
511	No version found
512	Incomplete object
513	Incomplete attribute
514	Incomplete NLIST
515	Invalid repeat count

This section describes the import leases and export leases file format.

The syntax is:

field1|field2|field3|...

The fields are listed next. If, in the import file, you chose not to supply the information for an optional field, you need to use delimiters ( | ) so that the number of fields is still 12. For example, type xyz|abc||123:

MAC address in xx:xx:xx:....:xx format (required)

MAC address type (required)

MAC address length (required)

IP address in dotted decimal format, a.b.c.d (required)

Start of lease time (GMT) (optional)

Lease expiration time (GMT) (optional)

Allowable extension time (GMT) (optional)

Last transaction time (GMT) (optional)

IP address of the DHCP server (optional)

Host name (without domain) (optional)

width=19 border=0>Domain name (optional)

Client ID (optional)

VPN (optional)

---

**Note:** For all the time fields, you can use either the number of seconds since 1970, or day, month, date, time, year format (Mon Apr 13 16:35:48 1998).

---

All contents are Copyright © 1992--2022 Cisco Systems, Inc. All rights reserved.



**AT\_AMELST** (amelist)

A list of DNS Address Match elements. For example: '10.10.0.0/16, lab-acl'. The usage follows BIND conventions. See the BIND 9 documentation for `acl` and `address_match_element`.

**AT\_ARRAY** (array)

A sequence of strings. This attribute type is deprecated in favor of the more general `AT_NLIST` type for lists of all other types.

**AT\_ATTRTYPE** (attrtype)

A CNR attribute type, stored as an integer.

**AT\_BITARRAY** (bitarray)

A bit-array, stored as an `AT_BLOB` of a fixed size.

**AT\_BLOB** (blob)

A sequence of unsigned octets. For example: '02:00:0a:00'.

**AT\_BOOL** (bool)

An 8-bit boolean value with only 0 and 1 as the legal values.

**AT\_BYTE** (byte)

An unsigned 8 bit integer value.

**AT\_CALCBIT** (calcbit)

A specialized type used to in representing certain DHCP options, such as option-81, the 'client-fqdn' option.

**AT\_CALCFLAG** (calcflag)

A value whose type is defined by the byte value that precedes it. Example: `dhcp-v4 option-122, suboption-3`. The extra-value field contains a map between the flag-byte and the `AT_XXX` type to use for parse/unparse.

**AT\_CIS** (cis)

A string that has case-insensitive comparison properties.

**AT\_CLEARTEXT** (clrtxt)

A string attribute that contains the clear text of sensitive information, such as a password.

**AT\_CONTAINER6** (container6)

A sequence of octets representing a DHCPv6 option that has encapsulated options.

**AT\_DATE** (date)

A 32-bit integer value representing a point in time to 1 second resolution. Format is 'Month Date hh:mm:ss Year'. For example: 'Jun 05 00:00:00 1980'.

**AT\_DICT** (dict)

Deprecated

**AT\_DNSNAME** (dname)

A fully qualified DNS name, encoded in DNS wire format with counted labels.

**AT\_DNSTTL** (dnsttl)

A signed integer with the semantics of a DNS ttl; only -1 is allowed as a negative number, with the special meaning 'use the zone default'.

**AT\_ENCRYPT** (encdata)

A sequence of bytes holding an encrypted form of sensitive information, such as a password.

**AT\_ENUMBYTE** (enumbyt)

An 8-bit integer with a fixed set of valid values that have an associated string name.

**AT\_ENUMINT** (enumint)

A 32-bit integer with a fixed set of valid values that have an associated string name.

**AT\_ENUMSHORT** (enumshort)

A 16-bit integer with a fixed set of valid values that have an associated string name.

**AT\_ENUMSTR** (enumstr)

A string with a fixed set of valid values.

**AT\_ESTRING** (estr)

Deprecated.

**AT\_EXPR** (expr)

An embedded expression with LISP-style syntax. For example: '(concat prefix var suffix)'. This expression is evaluated based on the context to produce a typed value.

**AT\_FILESIZE** (filesz)

An unsigned 64-bit integer representing the byte size.

**AT\_FILTER** (filter)

Deprecated.

**AT\_FLAGSINT** (flags)

A 32-bit integer with distinguished names associated with each bit position.

**AT\_GENADDR** (genaddr)

Deprecated.

**AT\_IFNAME** (ifname)

Deprecated.

**AT\_INT** (int)

An unsigned 32-bit integer.

**AT\_INT100** (int100)

An unsigned 32-bit integer number of 1/100's value. This is used for storing percentages in integer form.

**AT\_INT64** (int64)

An unsigned 64-bit integer.

**AT\_INT8** (decimal-byte)

A one-octet value with the same semantics as [AT\\_INT](#).

**AT\_INTI** (inti)

An unsigned 32-bit integer in Intel byte order.

**AT\_IP6** (ip6)

A 17-octet sequence representing an IPv6 address or prefix. It consists of 16 octets of address followed by an octet of 255, or 16 octets of prefix (with bits beyond the prefix-length being 0) followed by an octet of the prefix-length (0-128).

**AT\_IP6ADDR** (ip6addr)

A 128-bit IPv6 address.

**AT\_IP6NET** (ip6net)

A 17-octet sequence representing an IPv6 address or address with prefix-length. It consists of 16 octets of address followed by an octet of 255 (for address) or the prefix-length (0-128).

**AT\_IPADDR** (ipaddr)

A 32-bit IPv4 address.

**AT\_IPKEY** (ipkey)

An IP address that can be associated with a port number and/or a required TSIG key name.

The canonical textual values are:

- [<address>](#)
- [<address>:<port>](#)
- [<address>:<port>-<key>](#)
- [<address>-<key>](#)

**AT\_IPNET** (net)

An IPv4 address and a count of the bits that comprise the network number.

**AT\_IPPAIR** (ippair)

A pair of IPv4 addresses, stored in 8 octets.

**AT\_KEY** (key)

A sequence of bytes holding a shared-secret key that has a base-64 string representation.

**AT\_LISTREF** (listref)

Deprecated.

**AT\_MACADDR** (macaddr)

A MAC address, most frequently a 6 byte ethernet address with type 1, but more generally an arbitrary 1 byte type id, a 1 byte length and then 'length' address bytes. For example: '1,6,aa:bb:cc:dd:ee:ff'.

**AT\_MASK** (mask)

An [AT\\_IPADDR](#) that is in the form of an IP address mask (its binary sequence matches 1\*0\*).

**AT\_MESSAGE** (dhcpmsg)

A DHCP message type.

**AT\_MSDATE** (msdate)

A 64-bit value that represents a specific point in time to millisecond resolution.

**AT\_MSG6** (msg6)

A sequence of octets representing a DHCPv6 message in wire format.

**AT\_MSTIME** (mstime)

A 64-bit integer value that represents a span of time to millisecond resolution.

**AT\_MULTI** (multi)

A type that may contain data of multiple data types.

**AT\_NAMEREF** (nameref)

A string that refers to another object by name.

**AT\_NDICT** (ndict)

Deprecated.

**AT\_NLIST** (nlist)

A list of elements of some other type. This is the preferred type for storing lists.

**AT\_NODE** (dhcpnode)

Deprecated.

**AT\_NOLEN** (no length)

A DHCP option code with no length or value. For example: PAD or END.

**AT\_NSTRING** (nstr)

A string that is stored as a counted sequence, and is not necessarily null-terminated.

**AT\_OBJ** (obj)

A CNR object of any schema class.

**AT\_OBJREF** (objref)

A reference to a specific class of object by OID. It is similar to an [AT\\_OID](#), but adds the additional expectation that the referenced object has the specified class and does exist in the database.

**AT\_OID** (oid)

An 8-byte object id. The AT\_OID type differs from the [AT\\_OBJREF](#) in that it does not imply that the OID can be resolved to any specific type of object.

**AT\_OPTION** (option)

A sequence of octets representing a DHCPv4 option value in wire format.

**AT\_OPTION6** (option6)

A sequence of octets representing a DHCPv6 option value in wire format.

**AT\_OPTIONID4** (optionid4)

A DHCPv4 option number, stored as a four-octet integer.

**AT\_OPTIONID6** (optionid6)

A DHCPv6 option number, stored as a four-octet integer.

**AT\_OVERLOAD** (overld)

A type representing the DHCP option-overload option as a single octet.

**AT\_PACK** (pack)

An [AT\\_BLOB](#) that has some kind of structure associated with it.

**AT\_PAD** (dhcppad)

The DHCP PAD option, a single zero octet.

**AT\_PCV** (pcv)

A 32 bit product compatibility version number. The components of this number are (from high to low): major: 8 bits, minor: 8 bits, revision: 16 bits.

**AT\_PERCENT** (percent)

An integer bounded to the range 0-100, with a normal ascii output form that ends with a '%'. The value is stored as a single byte.

**AT\_PREFIX** (prefix)

A 17-byte sequence representing an IPv6 prefix. It consists of 16 bytes of prefix (with bits beyond the prefix-length being 0) followed by 1-byte of prefix-length (0-128). This is like [AT\\_SUBNET](#) for IPv6.

**AT\_PRIORITY** (priority)

A positive number where the precedence order is 1-n, followed by 0.

**AT\_RANGE** (range)

A 64-bit value containing a pair of 32-bit integers defining a range of integer values. This differs from the bounded integer and bounded time types.

**AT\_RANGEBYTE** (rangebyte)

An [AT\\_BYTE](#) value that is restricted to a range of valid values.

**AT\_RANGEINT** (rangeint)

An [AT\\_INT](#) value that is restricted to a range of valid values.

**AT\_RANGESHORT** (rangeshort)

An [AT\\_SHORT](#) value that is restricted to a range of valid values.

**AT\_RANGETIME** (rangetime)

An [AT\\_TIME](#) value that has an associated range of valid values.

**AT\_RDNSNAME** (rdname)

A relative DNS name, encoded in DNS wire format with counted labels.

**AT\_REQUEST** (dhcpreq)

A DHCP REQUESTED\_OPTIONS option, which the DHCP client uses to request option data by option number.

**AT\_RETCODE** (retcode)

A status code or return code.

**AT\_ROF\_FQDN** (rel or full fqdn)

A relative or fully qualified dns name.

**AT\_RR** (rr pack)

DNS RR encoded in DNS wire format.

**AT\_RULE** (rule)

A rule string, used in a DNS update-policy.

**AT\_SBYTE** (sbyte)

A signed 8 bit integer value.

**AT\_SECRET** (secret)

An OID object reference to a Secret object that stores the actual secret.

**AT\_SET** (set)

Deprecated.

**AT\_SHORT** (short)

An unsigned 16-bit integer.

**AT\_SHORTPAIR** (shortpair)

A 32-bit value holding a pair of 16-bit unsigned integers that are restricted to a range of valid values.

**AT\_SHRTI** (shorti)

An unsigned 16 bit integer in Intel byte order.

**AT\_SINT** (sint)

A signed 32-bit integer.

**AT\_SINT64** (sint64)

A signed 64-bit integer.

**AT\_SINT8** (s-decimal-byte)

A one-octet value with the same semantics as [AT\\_SHORT](#).

**AT\_SINTI** (sinti)

A signed 32-bit integer in Intel byte order.

**AT\_SSHORT** (sshort)

A signed 16-bit integer.

**AT\_SSHRTI** (sshorti)

A signed 16 bit integer in Intel byte order.

**AT\_STIME** (stime)

A signed version of the [AT\\_TIME](#) type that allows negative time spans. The main use of this type is for time-zone offsets that may be positive or negative.

**AT\_STRING** (string)

A null-terminated sequence of ASCII bytes.

**AT\_STRUCT** (struct)

Deprecated.

**AT\_SUBNET** (subnet)

An IPv4 address and a count of the bits that comprise the network number. The address component will have its host bits set to 0. A similar type, [AT\\_IPNET](#) does not assume or require the host bits to be 0.

**AT\_TAG** (tag)

A case-insensitive, restricted character set string.

**AT\_TEXPR** (texpr)

A typed expression; its structure and representation is the same as an [AT\\_EXPR](#) but the extra value of the attribute contains the expected type of the value produced by evaluating this expression.

**AT\_TIME** (time)

An unsigned integer number of seconds. It differs from the [AT\\_DATE](#) type in that it encodes a span or duration of time rather than an exact point in time. Format is 'HH:MM:SS'.

**AT\_TLV** (tlv)

A generic type-length-value tuple, in which the 'type' and 'length' are each one octet long.

**AT\_TLV2** (tlv2)

A generic type-length-value tuple, in which the 'type' and 'length' are each two octets long.

**AT\_TLV4** (tlv4)

A generic type-length-value tuple, in which the 'type' and 'length' are each four octets long.

**AT\_TYPECNT** (type-cnt)

This is a special-purpose type used in configuring DHCP options used by PXE clients. It holds a repeating pattern of tuples, each consisting of a two-octet type, a length byte, and 'length' bytes of data.

**AT\_VENDOR\_CLASS** (vendor-class)

An [AT\\_BLOB](#) representing a DHCP vendor-class option. An enterprise ID is followed by opaque data. (If DHCPv4, the enterprise ID is followed by an EID length.)

**AT\_VENDOR\_NOLEN** (vendor-nolen)

An [AT\\_BLOB](#) representing a DHCP vendor-class option. An enterprise ID is followed by tuples of vendor-specific data. The enterprise ID is never followed by an ID-length.

**AT\_VENDOR\_OPTS** (vendor-opts)

An [AT\\_BLOB](#) representing DHCP vendor-specific options data. An enterprise-id that is followed by TLVs of vendor-specific data. If DHCPv4, the enterprise ID is followed by an EID length.

**AT\_VERSION** (version)

A two-component 32 bit number, with 16 bits of major version and 16 bits of minor version.

**AT\_VPNID** (vpnid)

A 7-byte standard VPN ID, as defined in RFC 2685.

**AT\_VPREFIX** (prefix-var)

A 1 to 17-octet sequence representing an IPv6 prefix. See RFC 7227, section 5.3.

**AT\_ZEROSIZE** (zero-size)

A sequence of bytes representing certain DHCP options that have an option code and a length, where the length is always zero.