Using the nrcmd Commands

This chapter contains descriptions for all the **nrcmd** commands and their attributes, alphabetically arranged in sections by command. The command syntax appears at the beginning of each section, followed by syntax descriptions, attribute descriptions, and any usage guidelines. See the "Attribute Flags" section on page 1-6 for the types of attributes—required, optional, and read-only.

Attributes with time values are described with a default unit of time. However, you can append the characters **s**, **m**, **h**, **d**, **w**, or **y** immediately after the time value to translate the unit into seconds, minutes, hours, days, weeks, or years, respectively, if it fits in the allowed range of values. You can also mix time units in a single value, for example the equivalent **1d6h** and **1d360m**.

Table 2-1 lists all the **nrcmd** commands.

dhcp-link-policy

Table 2-1 CLI Commands

Command	Command	Command	Command
acl	dhcp-prefix	lease	scope
address-block	dhcp-prefix-policy	lease6	scope-policy
addr-trap	dhcp-subnet	lease-notification	scope-selection-tag
admin	dns	license	scope-template
ccm	dns-interface	option	scope-template-policy
client	dns-update-map	option-set	server
client-class	exit	owner	session
client-class-policy	export	policy	snmp
client-policy	extension	quit	snmp-interface
cluster	failover-pair	region	subnet
dhcp	group	remote-dns	tftp
dhcp-address-block	ha-dns-pair	report	trap-recipient
dhcp-address-block-policy	help	role	update-policy
dhcp-dns-update	import	router	vpn
dhcp-interface	key	router-interface	zone
dhcp-link	ldap	router-type	zone-dist
		save	zone-template

acl

Use the **acl** command to create an access control list (ACL) with which you can specify who has access to perform DNS updates, query resource records, and perform zone transfers. You can include these items in the ACL:

- Transaction Signature (TSIG) keys
- IP addresses
- Network addresses (address and mask)
- Other ACLs

Use the **acl** command together with the *update-acl, restrict-xfer-acl*, *restrict-query-acl*, and *restrict-recursion-acl* attributes of the **dns** and **zone** commands. ACLs also come into play when configuring rules for DNS update policies (see the "update-policy" section on page 2-190).

Note Be careful to use existing named ACLs with these attributes. Otherwise, the DNS server starts up, but any reference to the ACL disables the specified action. For example, if the update-acl attribute is set to a named ACL on the zone example and that named ACL does not exist, all updates on the zone example are disabled. This is true even if update-acl consists of more than one value.

Also, be aware that the restrict-recursion-acl is available only with the **dns** command.

acl *name* **create "**[**!**][**key**] *value*[**,**...]**"**

acl *name* **delete**

acl *name* **set match-list="**[**!**][**key**] *value*[**,**...]**"**

acl *name* **unset match-list**

acl *name* **get match-list**

acl *name* **add** [**!**][**key**] *value*

acl *name* **remove** [**!**][**key**] *value*

acl *name* [**show**]

acl list

acl listnames

Syntax Description

acl *name* **create "**[**!**][**key**] *value*[**,**...]**"**

Creates an ACL based on a match list of comma-separated TSIG keys (preceded by **key**), hosts or network addresses, or other ACLs. Use the **!** symbol for negation.

```
nrcmd> acl security-acl create "key securitykey1"
nrcmd> acl neg-acl create !192.168.2.1/16
```

You assign each ACL a unique name. However, the following ACL names have special meanings and you cannot use them for regular ACL names:

- **any**—Anyone can perform a certain action
- **none**—No one can perform a certain action
- **localhost**—Any of the local host addresses can perform a certain action
- **localnets**—Any of the local networks can perform a certain action

acl *name* **delete**

Deletes the ACL.

acl *name* **set match-list="**[**!**][**key**] *value*[**,**...]**"**

Sets the match list for the ACL, which can consist of TSIG keys, IP addresses, network addresses, or other ACLs, separated by commas and enclosed in quotes. A key must be preceded by the keyword **key**, and IP addresses must be in *address*/*mask* format.

You can create a name reference to an ACL in the match list before actually creating the ACL. However, the ACL must exist before you start or reload the DNS server. If the match list contains a reference to a nonexistent ACL, then the server will not start.

acl *name* **unset match-list**

Unsets the match list for the ACL.

acl *name* **get match-list**

Gets the named match list for the ACL.

acl *name* **add** [**!**][**key**] *value*

Adds an element to the match list.

acl *name* **remove** [**!**]**key** *value*

Removes an element from the match list.

acl *name* [**show**]

Shows the values associated with a specified ACL.

acl list

Displays all ACLs and the values associated with them.

acl listnames

Displays only the names of ACLs.

Related Commands **dns**, **key**, **update-policy**, **zone**

address-block

The **address-block** command creates and sets attributes for network address blocks created in the Network Registrar Central Configuration Management (CCM) database. An address block is a contiguous range of IP address space, and can parent one or more subnets.

Note A CCM address block is not the same as a DHCP address block used for delegation to DHCP servers in virtual private network (VPN) and subnet allocation deployments. You manage these DHCP address blocks using the **dhcp-address-block** command.

address-block *address/mask* **create** [*attribute*=*value* ...]

address-block *address/mask* **delete**

address-block *address/mask* **set** *attribute*=*value* [*attribute*=*value* ...]

address-block *address/mask* **unset** *attribute*

address-block *address/mask* **get** *attribute*

address-block *address/mask* [**show**]

address-block list

address-block listnames

Syntax Description

See Table 2-2 on page 2-5 for the **address-block** command attributes and their descriptions.

address-block *address/mask* **create** [*attribute*=*value* ...]

Creates a CCM address block with a network address (in the *address/mask* format), and optionally adds attributes. The policy is the only required attribute, which defaults to *default* if omitted.

address-block *address/mask* **delete**

Deletes a CCM address block.

address-block *address/mask* **set** *attribute*=*value* [*attribute*=*value* ...]

Sets one or more attributes for the CCM address block.

```
address-block 10.1.0.0/27 set vpn-id=1
```

address-block *address/mask* **unset** *attribute*

Unsets an optional CCM address block attribute.

address-block *address/mask* **get** *attribute*

Gets the explicitly defined value for a CCM address block attribute.

address-block *address/mask* [**show**]

Shows the values of all attributes of the CCM address block.

address-block list

Lists all CCM address blocks and their attributes.

address-block listnames

Lists only the names of all CCM address blocks.

Attributes

Table 2-2 describes the **address-block** command attributes and their values and defaults, if any.

Table 2-2 address-block Command Attributes

Attribute	Usage	Description
address	create set get	IP address of the CCM address block, specified at creation, defining its address range. Use the **set** command to redefine the address. Required, no default.
description	set get unset	Description of the use of the CCM address block. Optional, no default.
forward-zone-name	set get unset	Name of the forward DNS zone associated with the address block. Optional, no default.
owner	set get unset	Name of the owner possibly used to limit access to the address block. Optional, no default.
parent	set get unset	Name of the parent address block of the CCM address block, if any. Optional, no default.
region	set get unset	Name of the region possibly used to limit access to the address block. Optional, no default.
report-state	set get unset	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 to display.
reverse-zone-name	set get unset	Name of the reverse DNS zone associated with the address block. Optional, no default.
sink	set get unset	Name of the address sink (destination), if a leaf block delegated to a lower level sink. You cannot further split delegated address blocks into child address blocks. Optional, no default.
source	set get unset	Name of the address source, if a top-level address block allocated from a higher level source. Optional, no default.
type	set get unset	Name of the defined type of address block, if to be associated with a scope template, scope-selection tag, or client-class. Optional, no default.

Table 2-2 address-block Command Attributes (continued)

Attribute	Usage	Description
vpn	set get unset	ID of the virtual private network (VPN) used to support multiple address spaces, such as in a managed VPN environment. Optional, no default.

Related Commands dhcp-address-block, subnet, owner

addr-trap

The **addr-trap** command configures values for the address threshold notifications that a DHCP server sends. Use the **addr-trap** command to set low-threshold and high-threshold values for free address levels.

addr-trap *name* **create** [*attribute*=*value* ....]

addr-trap *name* **delete**

addr-trap *name* **set** *attribute*=*value*

addr-trap *name* **unset** *attribute*=*value*

addr-trap *name* **get** *attribute*

addr-trap *name* [**show**]

addr-trap list

Syntax Description

See Table 2-3 on page 2-7 for the **addr-trap** command attributes and their descriptions.

addr-trap *name* **create** *attribute*=*value* ....

Creates an SNMP address trap to define at which low and high values to send notifications concerning free address levels and how free addresses are aggregated.

addr-trap *name* **delete**

Deletes the specified SNMP address trap.

addr-trap *name* **set** *attribute*=*value*

Sets threshold values for free addresses.

addr-trap *name* **unset** *attribute*=*value*

Unsets threshold values for free addresses.

addr-trap *name* **get** *attribute*

Gets the explicitly defined value of the specified attribute.

addr-trap *name* [**show**]

Shows the values of all attributes on the specified trap.

addr-trap list

Lists all address traps.

Attributes

Table 2-3 describes the **addr-trap** command attributes.

Table 2-3 addr-trap Command Attributes

Attribute	Usage	Description
enable	**enable** disable get	Whether an address trap is active. Optional, default enabled, which activates the trap.

Table 2-3 addr-trap Command Attributes (continued)

Attribute	Usage	Description
high-threshold	set get unset	Value at which the DHCP server sends a high-threshold notification, re-enabling the low-threshold attribute. This threshold is triggered by the percentage of free addresses that exceed the specified value. Optional, default 25%.
low-threshold	set get unset	Value at which the DHCP server sends a low-threshold notification, reenabling the high-threshold attribute. This threshold is triggered by the percentage of free addresses that are below this value. Optional, default 20%.
mode	set get unset	How scopes aggregate free addresses. The values are: • **scope** (default)—Each scope tracks its free-address levels independently of other scopes. • **network**—Scopes that are members of the same primary subnet aggregate their free-address levels. • **selection tags**—Scopes that are members of the same primary subnet and that have exactly matching selection tags aggregate their free-address levels.
name	create set get	Unique name for the address trap. Required during creation.

Related Commands scope, scope-template, trap-recipient

admin

The **admin** command configures administrators for the cluster. You can choose any string for the administrator's name. Network Registrar uses a password to authenticate each administrator.

admin *name* **create** [*attribute*=*value*]

admin *name* **delete**

admin *name* **enable** *attribute*

admin *name* **disable** *attribute*

admin *name* **set** *attribute*=*value*

admin *name* **unset** *attribute*

admin *name* **get** *attribute*

admin *name* **enterPassword**

admin *name* [**show**]

admin list

admin listnames

Syntax Description

See Table 2-4 on page 2-10 for the **admin** command attributes and their descriptions.

admin *name* **create** [*attribute*=*value*]

Creates an administrator (and optionally sets one or more attributes). If an entry already exists, the command overwrites it. To avoid exposing the password through the *password* attribute, use the **admin** *name* **enterPassword** command instead.

admin *name* **delete**

Deletes an administrator.

admin *name* **enable** *attribute*

Enables an attribute.

admin *name* **disable** *attribute*

Disables an attribute.

admin *name* **set** *attribute*=*value*

Sets an administrator attribute. To avoid exposing the password through the *password* attribute, use the **admin** *name* **enterPassword** command instead.

admin *name* **unset** *attribute*

Unsets an attribute.

admin *name* **get** *attribute*

Gets the explicitly defined value of the specified attribute. Passwords are displayed as asterisks (*).

admin *name* **enterPassword**

Returns entry and confirmation prompts for a password, which is not echoed on the screen.

admin *name* [**show**]

Shows an administrator name and attributes.

admin list

Lists all administrators and their attributes. Passwords are displayed as asterisks.

admin listnames

Lists just the administrators' names.

Attributes

Table 2-4 describes the **admin** command attributes and their values and defaults, if any.

Table 2-4 admin Command Attributes

Attribute	Usage	Description
groups	set get unset	List of administrator groups, separated by commas. This is usually set in the Network Registrar Web UI. However, you can edit the group list, based on the existing group names in the Web UI. Optional, no default.
password	set get unset	Administrator password. Using this attribute exposes the password as plain text. To avoid this, use the **admin** *name* **enterPassword** command. Optional, no default.
superuser	enable disable **unset**	Whether to give the administrator superuser privileges in the Web UI. There can be multiple superuser administrators in the Web UI. Optional, default unset, which is essentially disable.

Usage Guidelines

Adding an Administrator

Use the **admin** *name* **create** command to create an administrator and associated password. You can also determine the level of access to the Web UI and whether the administrator should be a superuser there. Enabling the *superuser* attribute creates a superuser in the Web UI, although creating this type of administrator should be severely limited. An additional *group* attribute is usually set in the Web UI, but you can edit this list in the CLI, based on the known groups in the Web UI. (The CLI does not check whether a group exists; if you reference a group, be sure to create it in the Web UI.)

Limited administrator access allows access to host, zone, and DHCP server settings in the Web UI through the CLI, but does not allow an administrator to create new users or view license key data. Full access allows additional access to the global administration functions.

Adding a Password Without Exposing It

Create an administrator and omit the password. See the "Adding an Administrator" section on page 2-10. Then, use the **admin** *name* **enterPassword** command to enter a password to prevent echoing it on the screen. You are prompted to verify the password:

```
nrcmd> admin bob create
nrcmd> admin bob enterPassword
password:
verify password:
```

Related Commands

group, **role**

ccm

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

ccm set *attribute*=*value* [*attribute*=*value*…]

ccm unset *attribute*

ccm get *attribute*

ccm [**show**]

Syntax Description

See Table 2-5 on page 2-12 for the **ccm** command attributes and their descriptions.

ccm set *attribute*=*value* [*attribute*=*value*...]

Sets one or more CCM server attributes and their values.

ccm unset *attribute*

Unsets a CCM server attribute value.

ccm get *attribute*

Gets the explicitly defined value of the specified CCM server attribute.

ccm [**show**]

Shows the CCM server attributes and their values.

Attributes

Table 2-5 describes the **ccm** command attributes and their values and defaults, if any.

Table 2-5 ccm Command Attributes

Attribute	Usage	Description
lease-history-detail	**enable** disable get	If polling for lease history data, causes CCM to ask for history detail data when polling DHCP servers and saves the detail data when it is returned. Optional, default enabled.
local-zone-edit-mode	set get unset	Default zone edit mode (staged or synchronous) to use for DNS updates by clients of the local cluster. Overridden by specific client requests. If unset, clients should always present the choice of mode to the user. Optional, default synchronous.
poll-lease-hist-interval	set get unset	How often to collect the lease history data from all DHCP servers. If set to 0, no polling occurs. Optional, default 4 hours.
poll-lease-hist-offset	set get unset	Fixed time of day for the lease history polling, interpreted as a time of day, with 0 being 12 midnight. The polling scheduler ensures that one of the polling events occurs at this time of day. For example, if you set the interval to 4 hours and the offset to 6am, the polling occurs at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, no default.
poll-lease-hist-retry	set get unset	How often to retry in the event of a failure to collect lease history data from a DHCP server. Optional, default 1 retry.

Table 2-5 ccm Command Attributes (continued)

Attribute	Usage	Description
poll-replica-interval	set get unset	Default value of how often to poll for configuration changes when replicating data from a local cluster. Optional, default 4 hours.
poll-replica-offset	set get unset	Offset (from 12 midnight) of when the polling events should occur, interpreted as a time of day, with 0 being 12 midnight. The scheduler for polling ensures that one of the polling events occurs at this time of day. For example, if you set the *poll-replica-interval* to 4 hours and the offset to 6am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, default 0 offset.
poll-subnet-util-interval	set get unset	How often to collect the subnet utilization from all the DHCP servers. If set to 0, then do not poll the data. Optional, default 4 hours.
poll-subnet-util-offset	set get unset	Fixed time of day for the subnet utilization polling, interpreted as a time of day, with 0 being 12 midnight. The polling scheduler ensures that one of the polling events occurs at this time of day. For example, if you set the interval to 4 hours and the offset to 6am, then the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, no default.
poll-subnet-util-retry	set get unset	How often to retry if data polling fails. Optional, default 1 retry.
poller-event-threads	set get unset	How many threads that the poller creates. Optional, default 1 thread.
regional-zone-edit-mode	set get unset	Default zone edit mode (staged or synchronous) to be used for DNS updates by clients at the regional cluster. Overridden if a mode is specified by a specific client request. If unset, clients should always present the choice of mode to the user. Optional, default staged.
scope-edit-mode	set get unset	Edit mode used for DHCP edits (staged or synchronous) by Web UI and CLI clients. If staged, edits are written to the database but are not immediately forwarded to the DHCP server (they remain unpublished by DHCP). If synchronous, edits are written to the database and are immediately forwarded for publishing to the DHCP server. Optional, default stage.
trim-changeset-age	set get unset	Minimum age that a change set object must be to be deleted by the change log trim function. Optional, default 1 year.
trim-lease-hist-age	set get unset	Minimum time to keep a lease history record in the database. Any lease history record older than this time is deleted when the next lease history database trimming operation occurs. To disable lease history trimming (which is not recommended, as the database will grow without bound), set *trim-lease-hist-interval* to zero. Optional, default 24 weeks.
trim-lease-hist-interval	set get unset	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 attribute is set to 0, the lease history database continues to grow without bounds. Optional, default 24 hours.

Table 2-5 ccm Command Attributes (continued)

Attribute	Usage	Description
trim-subnet-util-age	set get unset	Age to allow a subnet utilization trimming element to be before deciding to delete that element. If this value is set to zero, no trimming occurs. Optional, default 24 hours.
trim-subnet-util-interval	set get unset	How often to trim the old subnet utilization data. If set to 0, no automatic subnet utilization trimming occurs. Optional, default 24 hours.

Related Commands server

client

The **client** command assigns attributes to a specific client entry. These attributes determine what type of IP address or policy 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 client 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.

client {*name* | **default**} **create** [*attribute*=*value*…]

client {*name* | **default**} **delete**

client {*name* | **default**} **set** *attribute*=*value* [*attribute*=*value*…]

client {*name* | **default**} **unset** *attribute*

client {*name* | **default**} **get** *attribute*

client {*name* | **default**} [**show**]

client list

client listnames

Note In releases of Network Registrar prior to 6.2, the client name was restricted to the MAC address. To use the MAC address as the client name in this release, you must enable the DHCP server attribute *validate-client-name-as-mac*.

Syntax Description

See Table 2-6 on page 2-16 for the **client** command attribute descriptions.

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).

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

client *name* **delete**
client default delete

Deletes the client entry.

client *name* **set** *attribute=value* [*attribute=value…*]
client default set *attribute=value* [*attribute=value…*]

Sets one or more attributes for the client.

client *name* **unset** *attribute*
client default unset *attribute*

Unsets the value of an attribute for the client.

client *name* **get** *attribute*
client default get *attribute*

Gets the explicit value of an attribute for the client.

client *name* [**show**]
client default [**show**]

Shows the values of all attributes assigned to the client.

client list

Lists all clients and any attributes assigned to them.

client listnames

Lists just the client identifiers.

Attributes

Table 2-6 describes the **client** and **client-class** command attributes and their values and defaults, if any.

Table 2-6 client Command Attributes

Attribute	Usage	Description
action	set get unset	Clients and client-classes—Action to take for the client. Optional, default none. Use one or more of these comma-delimited tokens: • **exclude**—Server ignores all communication from this client. If you use the command on the default client (**client default set action=exclude**), only a client specifically registered through the **client** command can communicate with the server. • **one-shot**—Server does not renew or re-offer any lease made to the client (either directly or in a client-class entry). • **use-release-grace-period**—Server delays the effect of DHCPRELEASE messages that the client sends. A release-grace-period for the policy specifies the delay time. During the grace period, the client's lease is not available for any other client. • **none**—No action (the default).
add-to-environment-dictionary	set get unset	Clients and client-classes—Quoted and comma-separated string of attribute-value pairs added to the environment dictionary whenever the client or client-class is associated with an incoming request. Used to configure extensions or expressions without having to rewrite the executable code. If both the client and client-class values are set, the client-class value overwrites that of the client. Optional, no default.
authenticate-until	set get unset	Clients only—Limits the authentication time to the duration that you specify, in a 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. Optional, no default.

Table 2-6 client Command Attributes (continued)

Attribute	Usage	Description
client-class-name	set get unset	Clients only—Client-class to which the client belongs. If the client is not in a client-class, the DHCP server uses the **default** client-class. Optional, no default.
client-lookup-id	set get unset	Client-classes only—Expression that evaluates to a string (or a blob that is a valid string). The resulting value specifies the key value to be used to look up the client in the client database, either locally or through LDAP. Enclose a simple expression in double quotes, or prefix the pointer to the file containing the expression with the at symbol (@). Optional, no default.
default-vpn	set get unset	Clients and client-classes—VPN to put it in if it does not already have a *vpn-id* or *vrf-name* value. Optional, no default.
domain-name	set get unset	Clients and client-classes—Domain name of the zone to use when performing DNS updates. The server places the client's address (A) resource record in this zone. Optional, no default.
embedded-policy	get unset	Clients and client-classes—Embedded policy for the client, as set by the **client-policy** command. Read-only, but you can unset all the embedded policy attributes, while retaining the policy name.
host-name	set get unset	Clients and client-classes—String to replace or generate the client host name. The first form is a string that does not start with an at symbol (@). This form is used to override the DHCP client request *host-name* option value. When you enter a valid name, the DHCP server ignores the *host-name* option value and uses this attribute value instead. You can use any valid DNS name except that it cannot include underscores. The second form is a string that starts with the special token @. Network Registrar uses this form to signal special handling: • @*host-name-option*—The server uses whatever *host-name* DHCP option the client sends. This is the default behavior if there is no entry for the *host-name* option in either the client or client-class. • @*no-host-name-option*—The server drops the *host-name* DHCP option that the client sends and does not replace it. If you disable DNS name synthesis, the client has no name placed in DNS. • @*use-macaddress*—The server synthesizes 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. For the *host-name* string to have an effect, you must set **scope** *name* **set dynamic-dns=update-all** (the default) in the scope that includes the address. Optional, no default; if blank, uses the *host-name* DHCP option.
limitation-id	set get unset	Client-classes only—Expression that evaluates to a blob or a string that can be used as a blob. The resulting value relates leases for which there is a maximum limit on the number of simultaneous active ones allowed. Configure the limit using the **policy** *name* **set limitation-count** command. Also, see the *over-limit-client-class-name* attribute. Enclose a simple expression in double quotes, or prefix the pointer to the file containing the expression with the @ symbol. Optional, no default.

Table 2-6 client Command Attributes (continued)

Attribute	Usage	Description
over-limit-client-class-name	set get unset	Clients and client-classes—Client-class name to use if the client exceeds the allowable limit of simultaneous active leases with a common limitation ID (see the *limitation-id* attribute). Optional, no default.
override-vpn	set get unset	Clients and client-classes—VPN to put it in, no matter what it provides for a *vpn-id* or *vrf-name* value. If you specify an override VPN on the client-class, and a default VPN for the client, the override VPN on the client-class takes precedence over the default VPN on the client. Optional, no default.
policy-name	set get unset	Clients and client-classes—Policy to add to the Network Registrar DHCP policy search list for this client. Optional, no default.
selection-criteria	set get unset	Clients and client-classes—Scope-selection tag or (comma-separated) tags to build the scope inclusion list. See the "scope-selection-tag" section on page 2-158 for how to create scope-selection tags. Optional, no default.
selection-criteria-excluded	set get unset	Clients and client-classes—Scope-selection tag or (comma-separated) tags to exclude when building the scope exclusion list. See the "scope-selection-tag" section on page 2-158 for how to create scope-selection tags. Optional, no default.
unauthenticated-client-class-name	set get unset	Clients only—Name of the client-class to use if the client is no longer authenticated. Optional, no default.
user-defined	set get unset	Clients and client-classes—User-defined string, such as a foreign key in a separate authorization database. This attribute has no effect on server operation. Optional, no default.

Related Commands **client-policy**, **client-class**, **client-class-policy**, **policy**, **scope**, **scope-policy**, **scope-selection-tag**

client-class

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.

client-class *name* **create** [*attribute*=*value*…]

client-class *name* **delete**

client-class *name* **set** *attribute*=*value* [*attribute*=*value*…]

client-class *name* **unset** *attribute*

client-class *name* **get** *attribute*

client-class *name* [**show**]

client-class list

client-class listnames

Note You must enable client-class processing for the server for Network Registrar to recognize client-classes.

```
nrcmd> dhcp enable client-class
```

Syntax Description

See Table 2-6 on page 2-16 for the **client** command attribute descriptions. Except where noted in the table, many **client** command attributes also apply to the **client-class** command.

client-class *name* **create** [*attribute*=*value*…]

Creates the client-class (and optionally defines its attributes). You must enable client-class processing for this to go into effect.

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

client-class *name* **delete**

Deletes the client-class.

client-class *name* **set** *attribute*=*value* [*attribute*=*value*…]

Sets one or more attributes for the client-class. See Table 2-6 on page 2-16 for the attributes.

client-class *name* **unset** *attribute*

Unsets a client-class attribute.

client-class *name* **get** *attribute*

Gets the explicitly defined value for the specified client-class.

client-class *name* [**show**]

Shows the values of all attributes assigned to the client-class.

client-class list

Lists all client-classes and any attributes assigned to them.

client-class listnames

Lists just the client-class names.

Attributes

See Table 2-6 on page 2-16 for the attribute descriptions.

Related Commands

client, **client-policy**, **client-class-policy**, **dhcp**, **ldap**, **policy**, **scope-policy**

client-class-policy

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. 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.

Syntax Description

For the syntax and descriptions, see the "policy" section on page 2-130.

Attributes

See Table 2-34 on page 2-133 for the attribute descriptions. Except where noted in the table, many **policy** command attributes also apply to client-class policies.

Related Commands

client, **client-policy**, **client-class**, **policy**, **scope-policy**

client-policy

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.

Syntax Description For the syntax and descriptions, see the "policy" section on page 2-130.

Attributes See Table 2-34 on page 2-133 for the attribute descriptions.

Related Commands **client-class**, **client-class-policy**, **policy**, **scope-policy**

cluster

The **cluster** command configures a regional or local cluster.

cluster *name* **create** *ipaddress* [*attribute*=*value*…]

cluster *name* **delete**

cluster *name* **enable** *attribute*

cluster *name* **disable** *attribute*

cluster *name* **set** *attribute*=*value* [*attribute*=*value*…]

cluster *name* **unset** *attribute*

cluster *name* **get** *attribute*

cluster *name* [**show**]

cluster list

cluster listnames

Syntax Description

See Table 2-7 on page 2-24 for the **cluster** command attribute descriptions.

cluster *name* **create** [*attribute*=*value*…]

Creates the cluster, and optionally sets attributes.

cluster *name* **delete**

Deletes the cluster.

cluster *name* **enable** *attribute*

Enables a cluster attribute.

cluster *name* **disable** *attribute*

Disables a cluster attribute.

cluster *name* **set** *attribute*=*value* [*attribute*=*value*…]

Sets one or more attributes for the cluster.

cluster *name* **unset** *attribute*

Unsets a cluster attribute.

cluster *name* **get** *attribute*

Gets an explicitly defined value for the specified cluster.

cluster *name* [**show**]

Shows the values of all attributes assigned to the cluster.

cluster list

Lists all clusters and any attributes assigned to them.

cluster listnames

Lists just the cluster names.

Attributes

Table 2-7 describes the **cluster** command attributes and their values and defaults, if any.

Table 2-7 cluster Command Attributes

Attribute	Usage	Description
admin	set get unset	Administrator identity to use when contacting this cluster. Optional, no default.
cluster-id	set get unset	ID of the local cluster that is the authoritative source for this object. Set as part of replica data propagation. Optional, no default.
fqdn	set get unset	DNS name of the cluster server, not used for contacting the cluster. Optional, no default.
http-port	set get unset	HTTP port to use for non-SSL-secured connections to the web server for this cluster. Optional, no default.
https-port	set get unset	HTTPS port to use for SSL-secured connections to the web server for this cluster. This port is only used if the value of the *use-https-port* attribute is enabled. Optional, no default.
ipaddr	create set get	IP address of the cluster server. Used instead of the *fqdn* when making connections to the cluster. Required at creation, no default.
local-servers	set get unset	Transient list of servers associated with this cluster. Provided to make 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. Optional, no default.
name	create set get	Name of the cluster. Required at creation, no default.
password	set get unset	Password to use to authenticate the identity stored in the *admin* attribute. This clear-text value should not be used except within process memory. The corresponding *password-secret* should be used instead. Optional, no default.
password-secret	set get unset	ID of the secret representing the password used to authenticate the identity stored in the *admin* attribute. Optional, no default.
poll-lease-hist-interval	set get unset	How often to collect the lease history from the DHCP server for this cluster. If set to 0 then do not poll. Optional, no default.
poll-lease-hist-offset	set get unset	Fixed time of day for the lease history polling, interpreted as a time of day, with 0 being 12 midnight. The polling scheduler ensures that one of the polling events occurs at this time of day. For example, if you set the interval to 4 hours and the offset to 6am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, no default.

Table 2-7 cluster Command Attributes (continued)

Attribute	Usage	Description
poll-lease-hist-retry	set get unset	How often to retry if it fails to poll the data. Optional, no default.
poll-replica-interval	set get unset	How often to poll this server for replica data. Optional, default 4 hours.
poll-replica-offset	set get unset	Fixed time of day for the lease history polling, interpreted as a time of day, with 0 being 12 midnight. The poll scheduler for polling ensures that one of the polling events occurs at this time of day. For example, if you set the interval to 4 hours and the offset to 6am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, default 4 hours.
poll-subnet-util-interval	set get unset	How often to collect the subnet utilization from DHCP server for this cluster. If set to 0, then do not poll. Optional, no default.
poll-subnet-util-offset	set get unset	Fixed time of day for the subnet utilization polling, interpreted as a time of day, with 0 being 12 midnight. The polling scheduler ensures that one of the polling events occurs at this time of day. For example, if you set the interval to 4 hours and the offset to 6am, the polling would occur at 2am, 6am, 10am, 2pm, 6pm and 10pm. Optional, no default.
poll-subnet-util-retry	set get unset	How often to retry if subnet utilization data polling fails. Optional, no default.
product-version	set get unset	Product version number of the cluster in major, minor revision form. This value is updated when the cluster is resynchronized. Optional, no default.
remote-id	set get unset	ID on the remote cluster that refers back to the local one. 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. Optional, no default.
replication-initialized	enable **disable** get	Enables or disables whether data replication has already been initialized on this cluster. Optional, default disabled.
restore-state	get	Indicates whether the cluster was deactivated or is in the process of being restored from the replica database. Read-only. Optional, no default.
scp-port	set get unset	Port number to use for SCP communications. Optional, no default.
scp-read-timeout	set get unset	Time limit for how long to wait for data when reading an SCP message from this cluster. Optional, default 20 minutes.
shared-secret	set get unset	ID of the secret that is shared between the server storing this object and the cluster that it represents. Used to generate single sign-on authentication tokens. Optional, no default.

Table 2-7 cluster Command Attributes (continued)

Attribute	Usage	Description
use-https-port	enable **disable** get	Enables or disables the HTTPS port used in making single sign-on connections to the cluster. If disabled, or if the https-port attribute is not set, then the *http-port* value is used. Optional, default disabled.
use-ssl	set get unset	Security mode to use (optional, required, or none) when connecting to this cluster. If optional, and there is a security library installed, the secure connection is tried. If required, the connection is not tried unless the connection can be secured (this requires the security library to be installed). If none, a secure connection is not tried. Optional, default optional.

Related Commands server

custom-option

The **custom-option** command creates and deletes custom DHCP options. You can also use this command to redefine any predefined DHCP options. If you delete this option, its definition returns to its original value.

Note This command applies to Network Registrar releases prior to 6.2 only. For 6.2, use the **option-set** command.

custom-option *name* **create** *number type* [**desc="***string***"**]

custom-option *name* **delete**

custom-option *name* **set** *attribute*=*value* [*attribute*=*value*…]

custom-option *name* **unset** *attribute*

custom-option *name* **get** *attribute*

custom-option *name* [**show**]

custom-option list

custom-option listnames

Syntax Description

custom-option *name* **create** *number type* [**desc="***string***"**]

Creates a custom option with a name, maps it to an option number, defines the data type, and optionally sets an attribute value. The positional attributes (in their correct order) are:

- *name*—Name of the custom option. Be consistent in naming the options in lowercase.
- *number*—Option number. Creating custom options that use the site-specific numbers 128 through 254 avoids conflicting with public DHCP options (see RFC2489). The DHCP options are listed in the appendices to the *Network Registrar User's Guide*. (Use the **dhcp-option list** command to list the predefined DHCP options.)
- *type*—Valid data type. Table 2-8 on page 2-28 lists the option data types. Optional, no default.
- **desc="***string***"**—Description string. Optional, no default. Includes quotation marks if there are spaces between words.

```
nrcmd> custom-option blue create 101 BYTE_ARRAY
```

This example creates a custom option that overlays the public time-offset option with a new definition:

```
nrcmd> custom-option green create 2 INT desc="Option green overlays time-offset"
```

Note As shown in the example, you can create custom options that override public DHCP or BOOTP options. Cisco Systems highly recommends not doing this. Also, do not give the custom option a name in the form **option-***number*, unless *number* is a numeric value from 1 through 254 that is not defined. The following entry generates a `duplicate object—option already exists` message:

```
nrcmd> custom-option option-192 create 192 INT
```

custom-option *name* **delete**

Deletes a custom option. If the custom option is an overlay of a public option, the option reverts to its previous definition.

custom-option *name* **set** *attribute*=*value* [*attribute*=*value*…]

Sets or resets one or more attributes for a custom option:

- **name**=*name*—Changes the name of the custom option, preferably in lowercase.
- **number**=*number*—Changes the option number. Numbers 128 through 254 are reserved for site-specific options. Optional, no default. The DHCP option numbers you should avoid using are listed in the appendices to the *Network Registrar User's Guide*.
- **type**=*type*—Changes the valid data type (see Table 2-8). Optional, no default.
- **desc**=*string*—Adds or changes the description string. Optional, no default.

```
nrcmd> custom-option blue set desc="this is an option called blue"
```

custom-option *name* **unset** *attribute*

Unsets the value of a custom option attribute.

custom-option *name* **get** *attribute*

Gets the explicitly defined value of an attribute for the specified custom option.

custom-option *name* [**show**]

Shows the attributes of a custom option.

custom-option list

Lists all custom options and any attributes assigned to them.

custom-option listnames

Lists just the names of the custom options.

Option Data Types

Table 2-8 lists the option data types that the **nrcmd** program supports.

Table 2-8 Option Data Types

Option Data Type	Type Name (Number)	Definition
boolean	BOOL (1)	TRUE or FALSE.
byte	BYTE (7)	8-bit unsigned integer.

Table 2-8 Option Data Types (continued)

Option Data Type	Type Name (Number)	Definition
byte array	BYTE_ARRAY (8)	Sequence of bytes represented in the form *xx*[:*xx*…] in which *x* is a hexadecimal character 0 through 9 or a through f. For example, to enter a series of four bytes containing the values 192, 168, 73 and 144, enter their hexadecimal values as **c0:a8:49:90**. Enter the ASCII string *ABCijk123* as **41:42:43:69:6a:6b:31:32:33**.
IP address	IPADDR (5)	IP address in the form of a.b.c.d.
IP address array	IPADDR_ARRAY (6)	Array of IP addresses.
signed array	INT_ARRAY (3)	Array of 32-bit signed integers.
signed integer	INT (2)	32-bit signed integer.
string	STRING (4)	ASCII text string.
unsigned array	UINT_ARRAY (12)	Array of 32-bit unsigned integers.
unsigned integer	UINT (11)	32-bit unsigned integer.
word	WORD (9)	16-bit unsigned integer.
word array	WORD_ARRAY (10)	Array of 16-bit unsigned integers.

Examples

List predefined DHCP options:

```
nrcmd> dhcp-option list
```

Create a custom option called red, mapped to number 100, of type IPADDR:

```
nrcmd> custom-option red create 100 IPADDR
```

Create a custom option called blue, mapped to number 101, that is an array of bytes:

```
nrcmd> custom-option blue create 101 BYTE_ARRAY
```

Give custom option blue a description:

```
nrcmd> custom-option blue set desc="This is another option called blue."
```

Create a custom option that overlays predefined option 51 (*dhcp-lease-time*):

```
nrcmd> custom-option green create 51 INT desc="Option green overlays dhcp-lease-time."
```

Revert option 51 to its original definition:

```
nrcmd> custom-option green delete
```

Redefine the *dhcp-lease-time* option to change its name:

```
nrcmd> custom-option create DHCPLeaseTime 51 UINT
```

Related Commands

option, **owner**

dhcp

The **dhcp** command configures the DHCP server in the cluster. Because there is only one DHCP server in a cluster, you do not need to reference the server by name.

dhcp enable *attribute*

dhcp disable *attribute*

dhcp set *attribute*=*value* [*attribute*=*value*…]

dhcp unset *attribute*

dhcp get *attribute*

dhcp [**show**]

dhcp limitationList *ipaddr* [*limitation-id*] **show**

dhcp attachExtension *extension-point extension-name* [*sequence-number*]

dhcp detachExtension *extension-point* [*sequence-number*]

dhcp listExtensions

dhcp setPartnerDown *partner-server* [*date*]

dhcp getRelatedServers *column-separator*=*string*

dhcp updateSMS [**all**]

dhcp getStats [**all** | {[**server**] [**failover**] [**dhcpv6**]} [**sample**]]

dhcp resetStats

dhcp serverLogs show

dhcp serverLogs nlogs=*value* **logsize=***value*

Note See the "server" section on page 2-168 for other server commands, including logging.

Syntax Description

See Table 2-9 on page 2-33 for the **dhcp** command attribute descriptions.

dhcp enable *attribute*

Enables an attribute for the DHCP server.

```
nrcmd> dhcp enable client-class
```

dhcp disable *attribute*

Disables an attribute for the DHCP server.

```
nrcmd> dhcp disable import-mode
```

dhcp set *attribute=value* [*attribute=value…*]

Sets one or more attributes for the DHCP server.

```
nrcmd> dhcp set failover-load-balancing-backup-pct=50
```

dhcp unset *attribute*

Unsets the value of a DHCP server attribute.

dhcp get *attribute*

Gets the explicitly defined value of an attribute for the DHCP server.

```
nrcmd> dhcp get max-dhcp-requests
```

dhcp [**show**]

Shows the values of the DHCP server attributes.

dhcp limitationList *ipaddr* [*limitation-id*] **show**

Determines the DHCP clients and their leases that are currently associated by a common *limitation-id* for the client (see the "client" section on page 2-15). This is useful when a DHCP client was 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" section on page 2-130). 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.

dhcp attachExtension *extension-point extension-name* [*sequence-number*]

Sets the specified extension point to call an extension. This example adds an extension named test to the extension point **post-packet-decode**:

```
nrcmd> dhcp attachExtension post-packet-decode test 1
```

If the extension point is already configured to call an extension, use the sequence number to specify the order in which Network Register is to execute the extensions (1, 2, 3, ...). If you omit a sequence number, Network Registrar overwrites the existing extension with the new value. See Table 2-11 on page 2-48 for descriptions of the *extension-point* values.

dhcp detachExtension *extension-point* [*sequence-number*]

Detaches extensions from an extension point. This example removes the test extension from the *post-packet-decode* extension point. Network Registrar removes the extension at the specified sequence number. If you omit the sequence number, Network Registrar removes the extension at sequence number 1.

```
nrcmd> dhcp detachExtension post-packet-decode test 1
```

dhcp listExtensions

Lists the currently configured extensions and their sequence numbers (if you configure multiple extensions) at each extension point. Cisco recommends that you run listExtensions for an extension point before attaching a new one. Check the results to ensure that the new extension has a different sequence number than an existing one.

dhcp setPartnerDown *partner-server* [*date*]

Notifies the DHCP server that its partner DHCP server is down and moves all appropriate scopes into the PARTNER-DOWN state. Optionally, you can specify the date and time when the partner was last known to operate. The default is the current date. This command is the equivalent of the **server dhcp setPartnerDown** command.

Caution Confirm that the partner server is completely down before issuing the **setPartnerDown** keyword. This command is ignored if failover communication with the partner succeeds.

dhcp getRelatedServers *column-separator=string*

Gets the status of the connection between the DHCP server and its DNS, LDAP, or failover servers. You can optionally specify that the report use *string* for separating columns. This command is the equivalent of the **server dhcp getRelatedServers** command.

dhcp updateSms [**all**]

Causes the DHCP server to perform System Management Server (SMS) network discovery. Optionally, including **all** sends out all leased addresses from the DHCP server to SMS. If you do not include this parameter, the server sends only those addresses leased since the last time you used this command. This command is the equivalent of the **server dhcp updateSMS** command.

dhcp getStats [**all** | {[**server**] [**failover**] [**dhcpv6**]} [**sample**]]

Displays the DHCP server statistics generated by the total counters since the last server restart. You can request four categories of statistics, with one qualifying keyword:

- **all**—Displays available statistics from all supported categories. Cannot be used with any other category.
- **server**—Displays all statistics available for the DHCP server. Can be combined with the failover and dhcpv6 categories.
- **failover**—Displays all statistics available for the failover server. Can be combined with the server and dhcpv6categoriescategory.
- **dhcpv6**—Displays all statistics for an IPv6 DHCP server. Can be combined with the server and failover categories.
- **sample**—If this keyword is used with one or more categories, displays the last snapshot taken of the counter values.

dhcp resetStats

Resets statistics counters to zero.

dhcp serverLogs show

Shows the number of log files configured and the maximum size of each.

dhcp serverLogs nlogs=*value* **logsize=***value*

Sets the two server logging parameters: the number of log files (nlogs) and the maximum size of each (logsize). When you use this command, you must specify one or both of the attributes. When you set the logsize, append a K to the value to signify thousands of units or append an M to signify millions of units. For example:

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

Note For these changes to take effect, save the changes and reload or restart the affected server.

Attributes

Table 2-9 describes the **dhcp** command attributes and their values and defaults, if any.

Table 2-9 dhcp Command Attributes

Attribute	Usage	Description
activity-summary-interval	set get unset	Time between activity summary log messages if enabled in the *activity-summary* setting in *log-settings*. Optional, default 5m (minutes).
addr-blocks-default-selection-tags	set get unset	Default selection tag (or list of tags) to be associated with incoming DHCP subnet allocation requests that do not contain any subnet name data. Optional, no default.
addr-blocks-use-client-affinity	**enable** disable unset	The DHCP server tries to allocate subnets to clients using DHCP address blocks that they already used. Disabling this attribute causes the server to supply subnets from any suitable DHCP address block, based on other selection data in the clients' messages. Optional, default enable.
addr-blocks-use-lan-segments	enable **disable** unset	Controls whether DHCP subnet allocation uses the *lan-segment* attribute when configured on DHCP address blocks. Optional, default disable.
addr-blocks-use-selection-tags	**enable** disable unset	Controls whether the server compares the incoming DHCP subnet allocation requests' subnet name data with each DHCP address block's selection tags. A DHCP address block is only considered if the two match. Optional, default enable.
map-user-class-id	set get unset	Controls use of the user-class-id option by the server. Values are: • 0 (none)—ignores the user class-id. This is the default value. • 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.
client-cache-count	set get unset	Specifies the specified maximum number of clients in the client cache. The DHCP server allocates the amount at startup and frees it up at shutdown. If you set the value to 0, client caching is disabled. Optional, default 1000 clients.
client-cache-ttl	set get unset	Time to live for an entry in the client cache, in seconds. The DHCP server replaces the entries in memory after this period. Optional, default 10s.
client-class	enable **disable** unset	Controls whether the DHCP server uses the client and client-class configuration objects to affect request processing. Optional, default disable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
client-class-lookup-id	set get unset	Expression to use to determine a client-class solely on data in an incoming DHCP client request. The expression must return a string that is the name of a currently configured client-class, otherwise the value of string <none> must be returned. Any return that is not a string containing the name of a currently configured client-class or <none> is considered an error. Enclose a simple expression in double quotes, or prefix the pointer to the file containing the expression with the @ symbol (see the *Network Registrar User's Guide*). Optional, no default.
cnr-5-0-upgraded	get	Shows whether the DHCP server was upgraded from Network Registrar 3.5 to 5.0. Read-only.
collect-addr-util-duration	set get unset	Maximum period, in hours, that the server maintains address utilization data. To disable address utilization data collection, either unset this parameter or set it to 0 (the default). Together the *collect-addr-util-duration* and *collect-addr-util-interval* attributes can impact memory usage in that each utilization snapshot is 68 bytes. For example, if there are 10 scopes, *collect-addr-util-duration* is set to 24h, *collect-addr-util-interval* is set to 1h, then the server collects 24 snapshots. To maintain address utilization data for each scope in this example, the calculation is 10x24x68, or 16 KB of memory. Optional, default 0 hours.
collect-addr-util-interval	set get unset	Frequency, in minutes or hours, that the DHCP server should maintain address utilization data snapshots. Ignored if *collect-addr-util-duration* is not configured or is set to 0. (See the *collect-addr-util-duration* description for how these attributes work together and with what memory impact.) Optional, default 15m.
collect-performance-statistics	enable **disable** unset	Controls whether the DHCP server collects statistics for performance monitoring. Optional, default disable.
collect-sample-counters	enable **disable** unset	Controls whether the DHCP server collects activity-summary counters independently of the *log-settings* attribute flag setting (see Table 2-10 on page 2-46). Optional, default disable.
default-free-address-config	set get unset	Default free-address SNMP trap configuration for the server, used by all scopes that are not explicitly configured with a free-address trap. Optional, no default.
defer-lease-extensions	**enable** disable unset	Controls whether the server renews a client's lease that is less than halfway to its expiration. By default, the server defers the lease extension—does not renew the lease, but grants another one while keeping the lease period. This way, the server avoids extra database updates. However, if a client is more than halfway to expiration, this setting has no effect, and the server extends the lease to the full configured lease period. Optional, default enable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
delete-orphaned-leases	enable **disable** unset	Leases that are in the lease state database can be orphaned. When the DHCP server initializes its cache from the lease state database, it expects every lease to match a configured scope. If the server finds a lease that does not match any configured scope, this property controls whether to delete that lease from the database or to ignore that entry (the default). In either case, the server cannot use the lease. Optional, default disable.
delete-orphaned-subnets	enable **disable** unset	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 in the server, or if the server cannot locate a parent DHCP address block that contains the subnet, the server uses this attribute to decide whether to keep the subnet entry in the state database (the default) or to delete it permanently. Optional, default disable.
dns-timeout	set get	Time, in milliseconds, that the DHCP server waits for a response before retrying a DNS update request. Required, default 60000 milliseconds (1 minute).
docsis-version-id-missing	set get unset	String (maximum 255 characters) that gets substituted with the **%@docsis-vers%** variable in the **policy** command's boot-file attribute. 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. Optional, no default.
drop-old-packets	set get unset	Time, in seconds, that a packet can age and still be processed. If the server is very busy, this could delay processing packets in the UDP input queue. The DHCP protocol lets clients retry packets that are not processed in a few seconds. Therefore, allowing the server to process packets that are older than a few seconds could increase the congestion. If the age of a packet is greater than the value of this attribute when the server processes it, the server drops the packet. Optional, default 4 seconds.
drop-packet-on-extension-failure	**enable** disable unset	Controls whether the server drops a packet (if possible) when it encounters a failure in an extension. Optional, default enable.
equal-priority-most-available	enable **disable** unset	By default, when multiple scopes have the same nonzero allocation priority (see the scope *allocation-priority* attribute), the scope with the least available addresses is used to allocate an address for a new client (if not in a limitation list). If *equal-priority-most-available* is enabled when multiple scopes have the same nonzero allocation priority, then the scope with the most available addresses is used to allocate an address for 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. Default disable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
expression-configuration-trace-level	set get unset	Trace level to use when configuring DHCP expressions. The range is from 0 through 10, 0 being the lowest amount of tracing and 10 the highest: • **0**—No additional tracing • **1**—No additional tracing • **2**—Failure retry (the default) • **3**—Function definitions • **4**—Function arguments • **5**—Variable lookups and literal details • **6**—Everything There is no performance penalty to specify a high *expression-configuration-trace-level*, as expressions are configured only when the server is started. Optional, default 2 (failure retry).
expression-trace-level	set get unset	Trace level to use when executing DHCP expressions. The range is from 0 through 10, 0 being no tracing and 10 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) There is considerable performance penalty to any setting other than 0, 1, or 2. 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. Optional, default 2.
extension-trace-level	set get unset	Default value of the extension trace level for every request object. You can override this value by setting the *extension-trace-level* in a user-written extension. Setting the level to 0 (the default) causes very little tracing. Setting the level to 3 causes considerable tracing. Optional, default 0.
failover-bulking	**enable** disable unset	With failover enabled, controls whether a failover bind update (BNDUPD) contains multiple lease state updates. Affects only the lease state updates that DHCP client activity generates. Optional, default enable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
failover-poll-interval	set get unset	With failover enabled, the polling interval of the failover partners (in seconds) to confirm network connectivity. Optional, default 15s.
failover-poll-timeout	set get unset	With failover enabled, the interval (in seconds) after which failover partners who cannot communicate know that they lost network connectivity. Optional, default 60s.
failover-recover	set get unset	With failover enabled, time at which the server performs initialization and goes into RECOVER state. If server A is running, server B uses this to ask for the state of server A. Dates can be in the form –2h (two hours ago, for example) or *month day hour***:***minute*[**:***second*] *year*. Optional, no default.
force-dns-updates	enable **disable** unset	Controls whether the DHCP server retries a DNS update whenever a client renews its lease, even if the server thinks that the update was already completed successfully. Optional, default disable. This attribute uses one of the following values: • the forward DnsUpdateConfig object (if configured) • the reverse DnsUpdateConfig object (if configured) • the default (or where appropriate the server's configured value or default value).
get-subnet-mask-from-policy	enable **disable** unset	Controls whether the DHCP server searches all relevant policies for a subnet mask option when constructing a response to send to a client. Normally, the DHCP server retains the subnet mask configured in the scope containing the base being granted to the DHCP client. Optional, default disable.
ha-dns-failover-timeout	set get unset	Maximum time period, in seconds, the DHCP server waits for replies from a DNS server before DHCP fails over to its partner. Applies only if you have configured DHCP to perform HA DNS updates. Optional, default 30s.
hardware-unicast	**enable** disable unset	Controls whether the DHCP server sends unicast rather than broadcast responses when a client indicates that it can accept a unicast. This attribute is only available on these operating systems: Solaris, Windows 2000, and Windows NT. Optional, default enable.
ignore-cisco-options	set get unset	Skips processing the Cisco-specific DHCP options specified by name in the comma-separated list. The allowable option names are **vpn-id** (185), **cisco-vpn-id** (221), and **cisco-subnet-allocation** (220). Use this attribute only if clients use the options for other purposes. Optional, no default.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
ignore-icmp-errors	**enable** disable unset	With this attribute enabled (the default), if you configured the DHCP server to send ICMP ECHO (ping-before-offer) 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. Optional, default enable.
ignore-requests-for-other-servers	enable disable unset	Controls whether to prevent the normal DHCP server response to client requests for other servers. Normally, if the DHCP server sees a client requesting a lease from another server for an address that this server is configured to control, it sets the lease to unavailable. However, some clients could send request packets with bad server ID options (rather than packets actually directed to other servers) that the server could wrongly interpret as the address being unavailable. You can enable this attribute to prevent this from occurring. Optional, no default.
import-mode	enable **disable** unset	Controls whether to have the DHCP server recognize only packets generated from the **import leases** command and to ignore all others. You can use this attribute if you want to update your DHCP server and prevent clients from receiving addresses during this period. Optional, default disable.
inhibit-busy-optimization	enable **disable** unset	Controls whether to prevent the server from using optimization to recover from periods of congestion. By default, the DHCP server determines that it is heavily loaded when the number of request packets reaches two-thirds of the total allocated. It 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. Optional, default disable.
initial-environment-dictionary	set get unset	Contains attribute-value pairs that initialize all environment dictionaries in the DHCP server. 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*". Optional, no default.
ip-history	enable **disable** unset	Controls whether to record data for the IP history database (see the *Network Registrar User's Guide*). Optional, default disable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
ip-history-detail	enable **disable** unset	Controls whether to record detailed data for the IP history database (see the *Network Registrar User's Guide*). Optional, default disable.
ip-history-max-age	set get unset	If *ip-history* is enabled, 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. Optional, default 4w.
last-transaction-time-granularity	set get unset	Time, in seconds, to guarantee that the last transaction time is accurate. Do not set this lower than the default of 60 seconds. For optimal performance, set it to a value that is greater than half of your lease interval. Optional, default 60s.
ldap-mode	set get unset	Determines the preference for using LDAP servers if multiple LDAP servers are configured. Optional, no default. There are two possible values: • **1—round-robin**—The DHCP server ignores the servers' 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.
log-settings	set get unset	Determines which events to log in the log files. See Table 2-10 on page 2-46. 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. Optional, the default flags are default, incoming-packets, and missing-options.
mac-address-only	enable **disable** unset	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. Optional, default disable. Use this attribute carefully. 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 MAC address.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
map-radius-class	set get unset	Use of the RADIUS class attribute, if present in a request's *relay-agent* option. Optional, default 0. The values are: • **0—none**—Ignore the RADIUS class (the default). • **1—map-as-tag**—Map the RADIUS class to the scope-selection tag (see the "scope-selection-tag" section on page 2-158). • **2—map-as-class**—Map the RADIUS class to the client-class. • **3—append-to-tags**—Append the RADIUS class to the scope-selection tag.
map-radius-pool-name	set get unset	Use of the RADIUS framed-pool attribute, if present in a *relay-agent* option. Optional, default 0. The values are: • **0—none**—Ignore the RADIUS pool name (the default). • **1—map-as-tag**—Map the RADIUS pool name to the scope-selection tag (see the "scope-selection-tag" section on page 2-158). • **2—map-as-class**—Map the RADIUS pool name to the client-class. • **3—append-to-tags**—Append the RADIUS pool name to the scope-selection tag.
map-user-class-id	set get unset	Determines the handling of user class-id. This attribute is global and is set for all DISCOVER packets. Optional, default 0. The values are: • **0—none**—Ignore the user class-id option (default). • **1—map-as-tag**—Map the user class-id option to the scope-selection tag (see the "scope-selection-tag" section on page 2-158). • **2—map-as-class**—Map the user class-id option to the client-class.
max-client-leases	set get	Maximum number of leases any DHCPv6 client is allowed to have associated with it on a link. Optional, default 200 leases.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
max-dhcp-requests	set get	Controls the number of buffers the DHCP server allocates for receiving packets from DHCP clients and failover partners. When enabling failover, allocate at least 150 buffers. Up to 1500 buffers could be reasonable for high capacity installations. When buffer size exceeds capacity, a burst of DHCP activity can clog the server with requests that become stale before they are processed. This results in an increasing processing load that can 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. Required, default 500 buffers. 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	set get	Number of buffers that the DHCP server allocates for responding to DHCP clients and communicating with failover partners. The number of buffers allocated should be at least two times the number allocated for the *max-dhcp-requests* attribute. As many as several thousand is reasonable in some installations. Required, default 1000 buffers (if failover is configured, the server configures additional responses).
max-dns-renaming-retries	set get	Number of times that the DHCP server can try to add a host in DNS even if it detects that the host's name is already present. This controls the number of times the DHCP server tries to modify a host's name to resolve a conflict on each failed update. Required, default 3 retries.
max-dns-retries	set get	Number of times that the server tries to send dynamic updates to a DNS server. Required, default 3 retries.
max-dns-ttl	set get	Time to live (TTL) ceiling, in seconds, for DNS records added through DNS updates. When the DHCP server adds a DNS record, it sets the TTL to less than one-third of the lease time (*dhcp-lease-time*), or the *max-dns-ttl* ceiling value. (If the lease time is greater than 3 * *max-dns-ttl*, then the *max-dns-ttl* is used; otherwise the lease time.) Note that the DNS record's effective TTL could actually be the zone's minimum TTL. A value of 0 is not recommended. Required, default 86400s (1d).
max-ping-packets	set get unset	Number of buffers that the server allocates for sending and receiving ICMP ping messages, if you use the **scope** *name* **enable ping-clients** command. See Table 2-41 on page 2-153. Required, default 500 buffers.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
max-waiting-packets	set get unset	Number of packets that can wait for processing for an 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. See the *dropped-waiting-packets* log setting attribute in Table 2-10 on page 2-46. It also drops duplicate packets (whose XID, client ID, and MAC address are the same as one already queued). Optional, default 6 packets.
mcd-blobs-per-bulk-read	set get unset	Number of binary large objects (blobs) for a bulk read. Use this attribute to tune DHCP start and reload times. Generally, a higher value results in faster server start and reload times, at the cost of using more memory. Optional, no default.
multicast-addresses	set get unset	Default multicast addresses to enable on interfaces. The ff02::1:2 address is required if any DHCPv6 clients are directly connected to the link associated with an interface. The ff05::1:3 address is the default multicast address used by relay agents when relaying DHCPv6 requests. Optional, defaults: ff02::1:2 and ff05::1:3.
one-lease-per-client	enable **disable** unset	Controls whether to have the DHCP server release any other leases that the client may have had on this server (on another network segment). Because the default behavior for the Network Registrar DHCP server is to store all the leases that a client obtains, this attribute ensures that the DHCP server stores only one lease, but a large performance issue might occur when it performs this check. A client might obtain a number of leases if a user with a laptop travels throughout the building and requests leases at different locations on the network. Optional, default disable.
priority-address-allocation	enable **disable** unset	If enabled and the scope's *allocation-priority* attribute is set, then the scopes are considered in the order of the allocation priority; if *allocation-priority* is unset, then the scope's subnet address becomes the allocation priority. If the scope's *allocate-first-available* attribute is enabled or unset, then it is in effect; if it is disabled, then the least recently used addresses in the scope get priority. This provides a way to enable priority address allocation for the entire DHCP server without having to configure it for every scope. When exercising this overall control of the address allocation, the actual priority of each scope depends only on its subnet address, which may or may not be what is desired. You can override the allocation priority for any individual scope by configuring the scope's *allocation-priority* directly even while *priority-address-allocation* is enabled for the server. Optional, default disable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
return-client-fqdn-if-asked	**enable** disable unset	Controls whether the system returns the *client-fqdn* option to the client in the outgoing packet if the client requests it in the parameter request list. Optional, default enable. If enable, the flags are always set in the option to 0x3 and the RCODE1 and RCODE2 to 255. Whatever string came in is sent back, even if the *use-client-fqdn* attribute is turned off and no matter what the actual name is or may ultimately be in DNS.
save-lease-renewal-time	enable **disable** unset	If set to true, the server saves the lease renewal time (the minimum time in which the client is expected to issue a lease renewal) as part of the lease in persistent memory. Optional, default disable.
skip-client-lookup	enable **disable** unset	If enabled, causes the DHCP server not to look up the client entry for client-class processing. If disabled (the default), the server looks up the client entry first. Optional, default disable.
sms-lease-interval	set get unset	Sets the time interval, in milliseconds, between sending addresses to the System Management Server (SMS). After you install a future release of Microsoft BackOffice Resource Kit (which contains an enhanced version of smsrsgen.dll), reduce this interval or set it to 0. Optional, default 1100 milliseconds.
sms-library-path	set get unset	Overrides the internal default value for the name of the SMS dll. The default is the empty string. If you specify an empty string, the system defaults to the internal server default of smsrsgen.dll. Optional, no default.
sms-network-discovery	set get unset	Causes the DHCP server to generate SMS network discovery records. To enable this attribute, set it to 1; to disable it, set it to 0 (the default). Use this attribute in conjunction with the **dhcp updateSms** command (see the "server" section on page 2-168). Optional, default 0.
sms-site-code	set get unset	Specifies the site code of the SMS server that receives discovery records when you issue the **updateSms** keyword. You must initialize this attribute to the appropriate SMS site code for the **updateSms** keyword to operate. See the "server" section on page 2-168. Optional, no default.
synthesize-reverse-zone	**enable** disable unset	Controls whether the DHCP server automatically generates the name of the reverse zone (in-addr.arpa) that is updated with PTR records. If this attribute is enabled and the scope does not have an explicit *dns-reverse-zone-name* attribute configured, the server uses the leased IP address and *dns-host-bytes* attribute on a scope to generate the reverse zone name. Optional, default enable.
traps-enabled	set get unset	Determines the traps that this server is configured to emit. Optional, no default.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
trim-host-name	**enable** disable unset	Controls whether the DHCP server trims the *host-name* string to the first period character (used to update DNS update records and to return the *host-name* option to clients). If this attribute is enabled, the *host-name* is truncated before the period. If disabled, the server retains the period characters in the *host-name*. Optional, default enable.
update-dns-for-bootp	**enable** disable unset	If the server replies to a BOOTP request and offers a lease from a scope that is configured for DNS updates, the DHCP server checks this attribute before beginning the update. You can use this attribute to prevent DNS updates for BOOTP clients, while allowing updates for DHCP clients. Optional, default enable.
upgrade-unavailable-timeout	set get unset	Controls the time given a lease in the database that has no expiration, that is, it went unavailable before installing Network Registrar. Optional, default 86400s (1d).
use-client-fqdn	**enable** disable unset	Controls whether to examine the *client-fqdn* option for the host name. If there are characters after the first dot in a *client-fqdn* option, the server ignores them because it determines the domain from the scope. Set this attribute to false if you do not want the server to determine a host name from this option, possibly because the client is sending unexpected characters. Optional, default enable.
use-client-fqdn-first	**enable** disable unset	Controls whether to examine the *client-fqdn* option on incoming packets first, before the *host-name* option, when determining a host name for a client. If there is a *client-fqdn* option with a host name specified, the system uses that host name. If the system finds no *client-fqdn* option in the incoming packet, the system uses the *host-name* option. If this attribute is set to false, the system examines the *host-name* option first and uses any name found in that option. If that option does not appear, it examines the *client-fqdn* option for a host name. Optional, default enable.
use-dns-update-prereqs	**enable** disable unset	By default, the DHCP server uses prerequisites in its DNS update messages when it performs DNS updates on behalf of clients. If this parameter is set to false, the server does not include prerequisites. Without them, the last client who uses a given domain name is associated with that name, even if another client is already associated with it. Optional, defaul
use-host-name	**enable** disable unset	Controls whether to examine the *host-name* option. Disable this attribute if you do not want the server to determine a host name from this option, possibly because the client is sending unexpected characters. Optional, default enable.
use-ldap-client-data	enable **disable** unset	Controls whether the DHCP server attempts to read client-entry data using the configuration supplied by the **ldap** command. See the "ldap" section on page 2-108. Optional, default disable.

Table 2-9 dhcp Command Attributes (continued)

Attribute	Usage	Description
v6-client-class-lookup-id	set get unset	Expression to use to determine a client class solely on data contained in an incoming DHCPv6 client request. The expression must return a string which is the name of a currently configured client class, otherwise the string "<none>" must be returned. Any return that is not a string containing the name of a currently configured client class or "<none>" is considered an error. Optional, no default.
validate-client-name-as-mac	enable **disable** unset	If set, the user interfaces should require that the name of each client entry is a valid MAC address (or the literal string *default*) and should turn the name into the canonical MAC address format (1,6,*xx*:*xx*:*xx*:*xx*:*xx*:*xx*), which the DHCP server uses as the default client entry lookup key. If set to false, the user interfaces should allow creating client entries with arbitrary names, which could match the lookup keys generated from the *client-lookup-id* expression. Optional, default disabled.
version	get	Gets the current software version of the DHCP server. Read-only.
vpn-communication	**enable** disable unset	If enabled (the default), the DHCP server can communicate with DHCP clients on a different virtual private network (VPN) from that of the DHCP server by using an enhanced DHCP relay agent capability. This enhanced capability is signalled by the appearance of the *server-id-override* suboption in DHCP option 82. Optional, default enable.

DHCP Log Settings

See Table 2-10 for the log flags. The log settings enabled by default are *default*, *incoming-packets*, *and missing-options*.

You can modify the logging behavior of the DHCP server by setting flags on the *log-settings* attribute. For example, you can suppress warning messages for unconfigured or missing options:

```
nrcmd> dhcp set log-settings=default,incoming-packets
```

You can turn on client and client-class debugging for the DHCP server:

```
nrcmd> dhcp set log-settings=client-detail
```

Or, you can turn off debugging and only log default messages for the DHCP server. In each case, reload the server:

```
nrcmd> dhcp set log-settings=default
nrcmd> dhcp reload
```

Table 2-10 DHCP Log Flags

Flag	Messages Logged to name-dhcp-1-log
activity-summary	Activity summary counters. Useful when you enable many of the no-*xxx* log settings, because it provides some indication of the server activity without imposing the load required for a log message corresponding to each DHCP message. Note that enabling the DHCP *collect-sample-counters* attribute has the same effect. Logs every five minutes by default, which is the default setting of the *activity-summary-interval* attribute.
client-criteria-processing	When the server examines a scope to find an available lease or to determine if a lease is still acceptable for a client who already has one. This setting can be useful when configuring or debugging client-class scope criteria processing. It logs a moderate amount of data, so you should not leave it enabled for long.
client-detail	After every client-class client lookup operation. This line shows all the data found for the client as well as the data found in the client's client-class. This is useful when setting up a client-class configuration and for debugging problems in client-class processing.
default	At a low level in several parts of the DHCP server. This flag is on by default. If you reconfigure the default, this logging does not appear.
dns-update-detail	Additional log messages for all DNS operations. This flag is helpful in diagnosing problems in DNS update operations.
dropped-waiting-packets	When dropping packets due to the setting of the *max-waiting-packets* DHCP attribute. The server may drop packets if the queue length for any IP address exceeds the value of the *max-waiting-packets* attribute. If the *dropped-waiting-packets* attribute is enabled, the server logs a message whenever it drops a waiting packet from the queue for an IP address.
failover-detail	Failover protocol operations and state transitions. Setting this does not place a significant load on the server.
incoming-packets	As a single line for every incoming packet. This setting is especially useful when you initially configure a DHCP server or BOOTP relay, in that an immediate positive indication exists that the DHCP server receives packets.
incoming-packet-detail	With the contents of every DHCP packet received by the DHCP server in human readable form. This setting enables the built-in DHCP packet sniffer for input packets. The log files fill up (and turn over) very rapidly when you enable this setting. It also causes a significant performance impact on the DHCP server, so that you should not leave it enabled for long.
ldap-create-detail	When the DHCP server sends a request creating a lease state entry to an LDAP server, receives a response from an LDAP server, or retrieves a result or error message from an LDAP server.
ldap-query-detail	When the DHCP server initiates a query to an LDAP server, receives a response from an LDAP server, or retrieves a query result or an error message from an LDAP server.
ldap-update-detail	When the DHCP server sends a lease update request to an LDAP server, receives a response from an LDAP server, or a retrieves a result or error message from an LDAP server.

Table 2-10 DHCP Log Flags (continued)

Flag	Messages Logged to name-dhcp-1-log
leasequery	When processing leasequery packets without internal errors, and when a lease query results in an acknowledgement (ACK) or negative acknowledgement (NAK) message.
minimal-config-info	Reduces the number of configuration messages that Network Registrar logs when the server starts or reloads. In particular, the server does not log a message for every scope when this flag is set.
missing-options	When a policy does not include an option a DHCP client requests, so that the DHCP server cannot supply it.
no-dropped-bootp-packets	Prevents logging the single line message normally logged for every dropped BOOTP packet.
no-dropped-dhcp-packets	Prevents logging the single line message normally logged for every DHCP packet dropped due to DHCP configuration. See the *no-invalid-packets* flag for messages associated with packets dropped because they are invalid.
no-failover-activity	Prevents logging normal activity messages and some warning messages logged for failover. Serious error log messages continue to appear independently of this log setting.
no-failover-conflict	Prevents logging warnings about potential conflicts between failover partners, but still logs errors. Setting this log setting can greatly reduce the amount of logging produced by a failover without losing the errors.
no-invalid-packets	Prevents logging the single line message normally logged for every DHCP packet dropped for being invalid. See the *no-dropped-dhcp-packets* flag for messages associated with packets dropped because of the server configuration.
no-reduce-logging-when-busy	When the server is very busy. Normally, the server reduces logging when it becomes very busy, such as when it uses over two-thirds of the available receive buffers (which is itself a configurable value). To do this, it sets the *no-success-messages*, *no-dropped-dhcp-packet*, *no-dropped-bootp-packets*, *no-failover-activity*, and *no-invalid-packet* flags and clears everything else except the *activity-summary* flag. When it is no longer very busy, for example, when only one-third of the available receive buffers are used, the server restores the previous settings. Setting this flag prevents Network Registrar from taking these actions.
no-success-messages	Prevents logging the single line message normally logged for every successful outgoing DHCP response packet. This affects logging for only successful outgoing response packets, and can greatly increase server performance.
no-timeouts	Prevents logging messages associated with the timeout of leases or offers.
outgoing-packet-detail	Contents of every DHCP packet transmitted by the server in a human readable form. Enables the built-in DHCP packet sniffer for output packets. The log files fill up (and turn over) very rapidly when this setting is enabled. Enabling this setting also causes a performance impact on the server because of the volume of outgoing packets so you should not leave it enabled for long.
unknown-criteria	As a single-line when the DHCP server finds a client entry that specifies a *selection-criteria* or *selection-criteria-excluded* that is not found in any scope appropriate for that client's current network location.

Extension Points

Table 2-11 summarizes the extension points available for controlling the DHCP server (in general sequential order).

Table 2-11 dhcp Command Extension Points

Extension Point	Purpose
check-lease-acceptable	Reached immediately after the server determines that the current lease is acceptable for this client. The extension can examine the results of that operation and can cause the routine to return different results. **Caution** Use this extension point with extreme care. Incorrect usage can create an infinite loop in the server.
post-class-lookup	If called after the client-class lookup is performed, allows you to modify the values looked up, particularly the *limitation-id* attribute value.
post-client-lookup	Examines the results of the entire client-class processing operation and acts based on those results, such as rewriting the results or dropping the packet. Use this extension point to place data items in the environment dictionary to affect the processing of an extension running at the *pre-packet-encode* extension point. Note that you cannot change the client-class at this point, but you can override certain values determined by the client or client-class already examined.
post-packet-decode	First extension point encountered when a request arrives. It immediately follows the decoding of the input packet and precedes any processing on the data in the packet. The primary activity for an extension at this point is to read information from an input packet and act on it, for example, to rewrite the input packet.
post-send-packet	Updates an external process or database with data about a request or response.
pre-client-lookup	Runs only if you set **dhcp enable client-class** for the server. This extension point allows an extension to: • Modify the client that is looked up during client-class processing. • Specify individual data items to override any data items found from the client entry or the client-class that it specifies. • Instruct the server to skip the client lookup altogether. In this case, the only client data used is that specified. • Drop the packet.
pre-packet-encode	Rewrites information in the response packet that the DHCP server sends to the user. This extension point comes after the response packet is ready for encoding into a packet sent to the DHCP client. Typically, you can add options to the packet at this extension point. The server can also drop the packet at this point, but the server already recorded its values in its internal database.
pre-dns-add-forward	Chooses the name and affects the number of DNS retries during update operations. Network Registrar might call this extension point multiple times for a single DNS update operation.

Related Commands

dhcp-interface, **key**, **lease**, **policy**, **scope**, **server**

dhcp-address-block

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, an 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.

dhcp-address-block *name* **create** *address* [*attribute*=*value* ...]

dhcp-address-block *name* **delete**

dhcp-address-block *name* **enable** *attribute*

dhcp-address-block *name* **disable** *attribute*

dhcp-address-block *name* **set** *attribute*=*value* [*attribute*=*value* ...]

dhcp-address-block *name* **unset** *attribute*

dhcp-address-block *name* **get** *attribute*

dhcp-address-block *name* [**show**]

dhcp-address-block list

dhcp-address-block listnames

dhcp-address-block *name* **listsubnets**

Syntax Description

See Table 2-12 on page 2-50 for the **dhcp-address-block** command attributes and their descriptions.

dhcp-address-block *name* **create** *address* [*attribute*=*value* ...]

Creates a DHCP address block with a network address (in the *address*/*mask* format), and optionally adds attributes. The policy is the only required attribute, which defaults to *default* if omitted.

```
nrcmd> dhcp-address-block red create 10.1.0.0/16 policy=Policy1
```

dhcp-address-block *name* **delete**

Deletes a DHCP address block.

dhcp-address-block *name* **enable** *attribute*

Enables a DHCP address block attribute.

dhcp-address-block *name* **disable** *attribute*

Disables a DHCP address block attribute.

dhcp-address-block *name* **set** *attribute=value* [*attribute=value* ...]

Sets one or more attributes for the DHCP address block. The DHCP address block policy is the only required attribute, which defaults to the *default* policy if omitted.

```
nrcmd> dhcp-address-block red set vpn-id=1
```

dhcp-address-block *name* **unset** *attribute*

Unsets an optional DHCP address block attribute. You cannot unset the policy attribute.

dhcp-address-block *name* **get** *attribute*

Gets the explicitly defined value for a DHCP address block attribute.

dhcp-address-block *name* [**show**]

Shows the values of all attributes of the DHCP address block.

dhcp-address-block list

Lists all DHCP address blocks and their attributes.

dhcp-address-block listnames

Lists only the names of all DHCP address blocks.

dhcp-address-block *name* **listsubnets**

Lists the subnets created from the DHCP address block.

Attributes

Table 2-12 describes the **dhcp-address-block** command attributes and their values and defaults,

Table 2-12 dhcp-address-block Command Attributes

Attribute	Usage	Description
address	create set get	IP address of the DHCP address block, specified at creation. Use the **set** command to redefine the address. Required, no default.
default-subnet-size	set get unset	Default DHCP subnet size for allocations from this address. Optional, default 28 subnets.
deprecated	enable **disable** unset	Whether or not to deactivate the 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. Optional, default disable.
embedded-policy	get	Embedded policy object for this DHCP address block. Read-only. Gets its value from the **dhcp-address-block-policy** command (see the "dhcp-address-block-policy" section on page 2-52).
name	create set get	Name of the DHCP address block, specified at creation. Use the **set** command to redefine the name. Required, no default.

Table 2-12 dhcp-address-block Command Attributes (continued)

Attribute	Usage	Description
vpn	set get unset	Virtual attribute that you can set instead of the *vpn-id*. When you set the *vpn*, the ID of that VPN becomes the *vpn-id* attribute value. You can also get the *vpn* associated with the current *vpn-id*. Optional, no default.
vpn-id	set get unset	ID of the VPN in which the DHCP address block resides. You must define the VPN using the **vpn** *name* **create** *vpn-id* command. See the "owner" section on page 2-129. If unset, the global VPN is used. Optional, default is to use the current VPN set by the **session set current-vpn** command, or, if undefined there, no VPN. Optional, no default.
policy	set get	Name of the policy associated with the DHCP address block. See the "policy" section on page 2-130 to create the policy. Required, default is the default policy.
segment-name	set get unset	Label for the network that this DHCP address block is part of. 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. Optional, no default.
selection-tags	set get unset	List of tag strings that are compared with incoming allocation requests' selection tags. All of a request's tags must match a DHCP address block's selection tags so that the DHCP address block can be used to satisfy the request. Separate multiple tags with a comma (do not include commas in tag names). Optional, no default.

Related Commands dhcp-subnet, dhcp-address-block-policy

dhcp-address-block-policy

The **dhcp-address-block-policy** command configures DHCP embedded policies for DHCP address blocks. A dhcp-address-block-policy is a policy object embedded within (and limited to) a dhcp-address-block object. Each DHCP address block may contain option data within its embedded policy, and may refer to a named policy with more option data, for example a router IP address. For the priority of what option data the server returns to a DHCP subnet, see the *Network Registrar User's Guide* for a description of the policy reply options.

The DHCP server implicitly creates and deletes embedded dhcp-address-block-policies when you create or delete the corresponding DHCP address blocks. You manipulate the dhcp-address-block-policy using the name of the corresponding DHCP address block.

Syntax Description

For the syntax and descriptions, see the "policy" section on page 2-130.

Attributes

See Table 2-34 on page 2-133 for the attribute descriptions. Except where noted in the table, many policy command attributes also apply to DHCP address block policies.

Related Commands

acl, **client-policy**, **client-class**, **client-class-policy**, **policy**, **scope**

dhcp-dns-update

The **dhcp-dns-update** command creates DNS update configurations for DHCP. These update configurations are referenced on DHCP policies to control the DNS update, performed by the DHCP server.

dhcp-dns-update *name* **create** *attribute*=*value* [*attribute*=*value...*]

dhcp-dns-update *name* **delete**

dhcp-dns-update *name* **enable** *attribute*

dhcp-dns-update *name* **disable** *attribute*

dhcp-dns-update *name* **set** *attribute*=*value* [*attribute*=*value...*]

dhcp-dns-update *name* **unset** *attribute*

dhcp-dns-update *name* **get** *attribute*

dhcp-dns-update *name* **show**

dhcp-dns-update list

dhcp-dns-update listnames

Syntax Description

Table 2-13 describes the **dhcp-dns-update** command attributes and their values and defaults, if any.

dhcp-dns-update *name* **create** *attribute*=*value* [*attribute*=*value...*]

Creates a DNS update configuration by name, and optionally adds attribute values.

dhcp-dns-update *name* **delete**

Deletes a DNS update configuration.

dhcp-dns-update *name* **enable** *attribute*

Enables an attribute on a DNS update configuration.

dhcp-dns-update *name* **disable** *attribute*

Disables an attribute on a DNS update configuration.

dhcp-dns-update *name* **set** *attribute=value*

Sets the attributes on the DNS update configuration.

dhcp-dns-update *name* **unset** *attribute*

Unsets the value assigned to the specified attribute.

dhcp-dns-update *name* **get** *attribute=value*

Gets the explicitly defined value of an attribute for the DNS update configuration.

dhcp-dns-update *name* **show**

Shows the values of all attributes assigned to the DNS update configuration.

dhcp-dns-update list

Lists all DNS update configurations and any attributes assigned to them.

dhcp-dns-update listnames

Lists just the DNS update configuration names.

Attributes

Table 2-13 describes the **dhcp-dns-update** command attributes and their values and defaults, if any.

Table 2-13 dhcp-dns-update Command Attributes

Attribute	Usage	Description
backup-server-addr	set get unset	Address of the backup DNS server to which DNS updates are sent if the server at *server-addr* is down. Optional, no default.
backup-server-key	set get unset	TSIG key used to process all DNS updates for *backup-server-addr*. Optional, no default.
dns-host-bytes	set get unset	Tells DHCP how many of the bytes in a lease's 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. Optional. If unset, the value is derived from the host-bytes in the subnet mask.
dynamic-dns	set get unset	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. The choices are update-none, update-all, update-fwd-only, and update-reverse-only. Optional, default update-all.
forward-zone-name	set get unset	Name of the DNS forward zone to which a DHCP client's host name (A record) should be added. Optional, no default.
reverse-zone-name	set get unset	Name of the DNS reverse (in-addr.arpa) zone that is updated with PTR record. If a *reverse-zone-name* is configured, DHCP always uses it. If the *synthesize-reverse-zone* is enabled, the DHCP server derives a reverse zone name from the lease IP address and *dns-host-bytes*. Optional, if not set the server uses the *synthesize-reverse-zone.*
server-addr	set get unset	Address of the DNS server to which DNS updates are sent. Optional, no default.
server-key	set get unset	TSIG key used to process all DNS updates for the *server-addr*. Optional, no default.
synthesize-name	**enable** disable get	Controls whether the DHCP server automatically creates DNS host names for DHCP clients who do not provide names. The server can synthesize unique names for clients based on the *synthetic-name-stem* attribute. Optional, default value is TRUE. This attribute uses one of the following values: • the forward DnsUpdateConfig object (if configured) • the reverse DnsUpdateConfig object (if configured) the default (or where appropriate the server's configured value or default value).

Table 2-13 dhcp-dns-update Command Attributes (continued)

Attribute	Usage	Description
synthetic-name-stem	set get unset	Stem of the default hostname to use if clients do not supply hostnames. Optional, default is DHCP. This attribute uses one of the following values: • the forward DnsUpdateConfig object (if configured) • the reverse DnsUpdateConfig object (if configured) the default (or where appropriate the server's configured value or default value).
update-dns-first	enable **disable** get	Controls whether the DNS server is updated before the lease is granted. Optional, default disabled.
update-dns-for-bootp	**enable** disable get	If the server is replying to a BOOTP request, and is offering a lease from a scope that is configured to perform DNS updates, it checks this attribute before beginning the DNS update. This feature allows an administrator to prevent DNS updates for BOOTP clients, while allowing updates for DHCP clients. Optional, default enabled.

Related Commands dhcp

dhcp-interface

The **dhcp-interface** command adds, removes, and lists Network Registrar DHCP interfaces. In Network Registrar, a DHCP interface is a logical representation of the hardware interface (for example a server's Ethernet or Token Ring network interface card) that the DHCP server uses. The DHCP server uses the configured address information to determine which interface to use to send and receive packets. When the DHCP server finds a match to an interface address, it selects that interface and all addresses on that interface.

dhcp-interface *name* **create** *attribute*=*value* [*attribute*=*value...*]

dhcp-interface *name* **delete**

dhcp-interface *name* **enable** *attribute*

dhcp-interface *name* **disable** *attribute*

dhcp-interface *name* **set** *attribute*=*value* [*attribute*=*value...*]

dhcp-interface *name* **unset** *attribute*

dhcp-interface *name* **get** *attribute*

dhcp-interface *name* [**show**]

dhcp-interface list

dhcp-interface listnames

Syntax Description

Table 2-14 describes the **dhcp-interface** command attributes and their values and defaults, if any.

dhcp-interface *name* **create** *attribute*=*value* [*attribute*=*value...*]

Creates a DHCP interface specification named by the IP address and network prefix bits of the physical interface. You can specify the mask bits as 24 or 16.

dhcp-interface *name* **delete**

Deletes a DHCP interface.

dhcp-interface *name* **enable** *attribute*

Enables an attribute on a DHCP interface.

dhcp-interface *name* **disable** *attribute*

Disables an attribute on a DHCP interface.

dhcp-interface *name* **set** *attribute*=*value*

Sets the attributes on the DHCP interface. The *ignore* attribute enables or disables the server to ignore the named. You can set this attribute to disable to temporarily disable a specific interface in a list. To change the interface address, delete and recreate the interface. Optional, no default.

dhcp-interface *name* **unset** *attribute*

Unsets the specified attribute on the DHCP interface.

dhcp-interface *name* **get** *attribute*=*value*

Gets the explicit value of an attribute for the DHCP interface.

dhcp-interface *name* [**show**]

Shows the values of all attributes assigned to the DHCP interface.

dhcp-interface list

Lists all DHCP interfaces and any attributes assigned to them.

dhcp-interface listnames

Lists just the DHCP interface names.

Attributes

Table 2-14 describes the **dhcp-interface** command attributes and their values and defaults, if any.

Table 2-14 dhcp-interface Command Attributes

Attribute	Usage	Description
address	set get unset	IP address and subnet mask of the interface or interfaces that the DHCP server uses. If you do not assign values, the interface is excluded from matching against the list of automatically discovered interfaces. Optional, no default.
ip6address	set get unset	IPv6 address and prefix length of the interface or interfaces that the DHCP server uses. If you do not assign values, the interface is excluded from matching against the list of automatically discovered interfaces. Optional, no default.
multicast	set get unset	The multicast addresses that you want to enable or disable on the DHCP interfaces. The defaults are ff02::1:2 and ff05::1:3. The address ff02::1:2 is required 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. Optional, defaults indicated.
name	create set get	Name of the interface that the DNS server uses. Required at creation, no default.

Usage Guidelines

Selecting Server Interfaces

By default, the DHCP server automatically uses all the network interfaces on your server. Use the **dhcp-interface** command to select specific interfaces. In Network Registrar, the interface name syntax is the IP address and subnet mask with the /*n* suffix, reflecting the number of bits in the network part of the address, or the IPv6 address and prefix. For example, a subnet mask in IP format 255.255.255.0 becomes a suffix of /24 (24 bits of network address); the IP mask 255.255.255.192 becomes a subnet mask suffix of /26, and so on. Be sure that both the address and subnet mask are accurate, using a utility like **ipconfig** in Windows, or **ifconfig** in Solaris/Linux.

If you delete the default interface (which is not advisable), the DHCP server uses hardcoded default values for port numbers and socket buffer sizes for the interfaces that it autodiscovers. You can also show and list interfaces, and unset or reset the address and mask values of a nondefault interface.

To have the DHCP server temporarily ignore an interface in a list of defined ones, use the **dhcp-interface** *address* **set ignore=true** command. If you enable the *discover-interfaces* attribute, the DHCP server consults the interface list for all defined interfaces with the *ignore* attribute set to *false*, and tries to listen on each of them.

Related Commands

dhcp

dhcp-link

The **dhcp-link** command configures IPv6 network links. Use links to group IPv6 prefixes (see the dhcp-prefix command) together.

dhcp-link *name* **create** [*attribute*=*value*]

dhcp-link *name* **delete**

dhcp-link list

dhcp-link listnames

dhcp-link *name* **show**

dhcp-link *name* **set** *attribute*=*value* [*attribute*=*value*...]

dhcp-link *name* **unset** *attribute*

dhcp-link *name* **get** *attribute*

dhcp-link *name* **enable** *attribute*

dhcp-link *name* **disable** *attribute*

Syntax Description

See Table 2-15 on page 2-59 for the **dhcp-link** command attribute descriptions.

dhcp-link *name* **create** [*attribute*=*value*]

Creates a link and optionally assigns attribute values. The name and attribute values are required for this command.

```
nrcmd> dhcp-link example-link create ff00::/8
```

dhcp-link *name* **delete**

Deletes a link.

dhcp-link list

Lists all links and any attributes assigned to them.

dhcp-link listnames

Lists the names of all links.

dhcp-link *name* **show**

Shows the values of all attributes assigned to a link.

dhcp-link *name* **set** *attribute*=*value* [*attribute*=*value* ...]

Sets an attribute to a value for a link.

```
nrcmd> dhcp-link example-pref set address_ff00::/10
```

dhcp-link *name* **unset** *attribute*

Unsets the value of a link attribute.

dhcp-link *name* **get** *attribute*

Gets the explicit value of an attribute for a prefix.

dhcp-link *name* **enable** *attribute*

Identifies the attributes that have been enabled on any specific link name.

dhcp-link *name* **disable** *attribute*

Enables the attribute for a prefix.

Attributes

Table 2-15 describes the **dhcp-link** command attributes and their values and defaults, if any.

Table 2-15 dhcp-link Command Attributes

Attribute	Usage	Description
description	set get unset	Describes the link. Optional, no default.
embedded-policy	set get unset	Policy embedded within a single specific link object for use when replying to clients. Optional, no default.
name	set get	User-assigned name for the link. Required at creation, no default.
policy	set get unset	Reference to a shared policy to use when replying to clients. Optional, no default.
vpn-id	set get unset	ID of the VPN containing the prefix. Optional, no default.

Related Commands

dhcp-prefix

dhcp-link-policy

Use the **dhcp-link-policy** command to configure a DHCP policy that is embedded in a DHCP dhcp-link. An embedded policy is a collection of DHCP option values and settings associated with another object, in this case a dhcp-link. A dhcp-link-policy is created implicitly when you first reference it, and is deleted when the address-block is deleted.

dhcp-link-policy *name* **delete**

dhcp-link-policy *name* **set** *attribute=value* [*attribute=value ...*]

dhcp-link-policy *name* **get** *attribute*

dhcp-link-policy *name* **disable** *attribute*

dhcp-link-policy *name* **enable** *attribute*

dhcp-link-policy *name* **show**

dhcp-link-policy *name* **setLeaseTime** *time-val*

dhcp-link-policy *name* **getLeaseTime**

dhcp-link-policy *name* **setOption** {*opt-name* | *id*}

dhcp-link-policy *name* **getOption** {*opt-name* | *id*}

dhcp-link-policy *name* **unsetOption** {*opt-name* | *id*}

dhcp-link-policy *name* **listOptions**

dhcp-link-policy *name* **setVendorOption** {*opt-name* | *id*} *opt-set-name value*

dhcp-link-policy *name* **getVendorOption** {*opt-name* | *id*} *opt-set-name value*

dhcp-link-policy *name* **unsetVendorOption** {*opt-name* | *id*} *opt-set-name value*

dhcp-link-policy *name* **listVendorOptions**

Attributes

Table 2-16 describes the **dhcp-link-policy** command attributes.

Table 2-16 dhcp-link-policy Command Attributes

Attribute	Usage	Description
affinity-period	set get unset	For DHCPv6, specifies for how long a lease that has become available is retained for a client before it is deleted. This allows a client to obtain an expired lease if the client returns during this period; or it can prevent a client from reusing an address if it returns within this period (if either inhibit-all-renews or inhibit-renews-at-reboot is enabled). Optional, no default.

Table 2-16 dhcp-link-policy Command Attributes (continued)

Attribute	Usage	Description
allow-client-a-record-update	enable **disable**	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. Optional, default is false.
allow-client-hints	enable **disable**	If allow-client-hints is true, addresses and prefixes requested by the client, in SOLICIT and REQUEST messages, are used if possible. If allow-client-hints is false, addresses and prefixes requested by the client are ignored. Optional, the default is false.
allow-dual-zone-dns-update	enable **disable**	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. Optional, the default is false.
allow-lease-time-override	enable **disable**	Indicates that clients may request a specific lease-time. The server will not honor those requested lease-times if this attribute is set to false. The server will not honor a client's lease-time if that time is longer than the server's normal lease-time. Optional, default is disabled.
allow-non-temporary-addresses	**enable** disable	Determines whether DHCPv6 clients can request non-temporary addresses. Optional, default is true.
allow-rapid-commit	enable **disable**	Determines whether DHCPv6 clients can use a 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. This attribute needs special handling in processing the policies. The server checks all prefix policies (both embedded and named) for the link to which the client has access: • 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. Optional, default is FALSE.
allow-temporary-addresses	**enable** disable	Determines wether DHCPv6 clients can request temporary addresses.
default-prefix-length	set get unset	For delegation, specifies the default length of the delegated prefix if it is not explicitly requested by the requesting router (client). The default length must always be less than or equal to the prefix length of the prefix range. Optional, default is 64.
forward-dnsupdate	set get unset	Specifies the forward zone DNS update. Optional, no default.
forward-zone-name	set get unset	Names an optional forward zone to update. Optional, no default.

Table 2-16 dhcp-link-policy Command Attributes (continued)

Attribute	Usage	Description
giaddr-as-server-id	enable **disable**	Enables the DHCP server to set the server-id option on a DHCPOFFER and a DHCPACK to the giaddr of the incoming packet, instead of the IP address of the server (as it will by default). 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 may not actually be an address to which the DHCP client can unicast a packet. In these cases, the DHCP client cannot renew a lease, and must always performing a rebind operation (where the DHCP client broadcasts a request instead of unicasting it to what it believes is the DHCP server). This feature is disabled by default.
grace-period	set get unset	Defines the length of time between the expiration of a lease and the time it is made available for re-assignment. Optional, default is 5m.
inhibit-all-renews	enable **disable**	Causes the server to reject all renewal requests, forcing the client to obtain a different address any time it contacts the DHCP server. Optional, default is false.
inhibit-renews-at-reboot	enable **disable**	Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot. Optional, default is false.
limitation-count	set get unset	Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active leases. Optional, no default.
longest-prefix-length	set get unset	For delegation, the longest allowable length for prefixes. If the requesting router (client) requests a prefix length that is longer than this, the value set in this attribute is used instead. Optional, the default is the default-prefix-length.
offer-timeout	set get unset	Tells the server to wait the specified amount of time if it has offered a lease to a client but the offer is not accepted. At the end of the specified time interval, the server makes the lease available again. Optional, default is 2m.
packet-file-name	set get unset	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. Optional, no default.
packet-server-name	set get unset	Identifies the host-name of a server to used 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. Optional, no default.
packet-siaddr	set get unset	Identifies the IP address of the next server in a client's 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. Optional, no default.

Table 2-16 dhcp-link-policy Command Attributes (continued)

Attribute	Usage	Description
permanent-leases	enable **disable**	Indicates that leases for this scope should be permanently granted to requesting clients. Optional, default is disabled.
preferred-lifetime	set get unset	Specifies the default and maximum preferred lifetime for leases to DHCPv6 clients. Optional, default value is 1w.
reverse-dnsupdate	set get unset	Specifies the reverse zone DNS update. Optional, no default.
server-lease-time	set get unset	Tells the server for how long a lease is valid. For more frequent communication with the client, it may be useful to have the server consider leases leased for a longer period than the client does. 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. Optional, no default.
shortest-prefix-length	set get unset	For delegation, The shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, the value set in this attribute is used instead. Optional, the default is the default-prefix-length.
split-lease-times	enable **disable**	If enabled, the DHCP server uses the value of the *server-lease-time* attribute internally. Clients are still offered lease times that reflect the configured lease-time option from the appropriate policy, but the server bases its decisions about expiration on the *server-lease-time* value. Optional, default is disabled.
unavailable-time out	set get unset	Permits the server to make a lease unavailable for the specified period of time and then to return the lease available state. Optional. 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	enable **disable**	When checking the server's database for IP addresses that reserved, the server by default uses the MAC address of the DHCP client as the key for the database lookup. If *use-client-id-for-reservations* is enabled, then the check for reserved leases is performed by using the client-id of the DHCP client. The client-id is usually supplied by the DHCP client. In cases where it was not supplied by the DHCP client, then it is synthesized by the server, and that value will be used. Optional, the default is disabled.
v4-bootp-reply-options	set get unset	Lists the options that are returned to all BOOTP clients, whether or not a client specifically asks for the option data. Optional, no default.
v4-reply-options	set get unset	Lists the options that are returned to all DHCPv4 clients, whether or not a client specifically asks for the option data. Optional, no default.
v6-reply-options	set get unset	A list of options that should be returned in any replies to DHCPv6 clients.

Table 2-16 dhcp-link-policy Command Attributes (continued)

Attribute	Usage	Description
valid-lifetime	set get unset	Specifies the default and maximum valid lifetime for leases to DHCPv6 clients. Optional, default value is 2w.

Usage Guidelines

You set individual option values with the **setOption** command and unset option values with the **unsetOption** command. To view option values, use the **getOption** command or the **listOptions** commands. When you set an option value, the DHCP server replaces any existing value or creates a new one as needed for the given option name.

dhcp-prefix

Use the **dhcp-prefix** command to configure IPv6 network prefixes. These prefixes configure DHCPv6 address allocation and prefix delegation.

dhcp-prefix *name* **create** *address* [*attribute=value*]

dhcp-prefix *name* **delete**

dhcp-prefix *name* **enable** *attribute*

dhcp-prefix *name* **disable** *attribute*

dhcp-prefix *name* **set** *attribute=value* [*attribute=value* ...]

dhcp-prefix *name* **unset** *attribute*

dhcp-prefix *name* **get** *attribute*

dhcp-prefix *name* **show**

dhcp-prefix list

dhcp-prefix listnames

dhcp-prefix *name* **listLeases**

dhcp-prefix *name* **addReservation** *ip6address* [*/prefix-length*] *duid*

dhcp-prefix *name* **removeReservation** {*ip6address* [*/prefix-length*] | *duid*}

dhcp-prefix *name* **listReservations**

Syntax Description

See Table 2-17 on page 2-66 for the **dhcp-prefix** command attribute descriptions.

dhcp-prefix *name* **create** *address*

Creates a prefix and optionally assigns attribute values. The name and address values are required for this command.

```
nrcmd> dhcp-prefix example-pref create ff00::/8
```

dhcp-prefix *name* **delete**

Deletes a prefix.

dhcp-prefix *name* **enable** *attribute*

Identifies the attributes that have been enabled on any specific prefix name.

dhcp-prefix *name* **disable** *attribute*

Enables the attribute for a prefix.

dhcp-prefix *name* **set** *attribute=value* [*attribute=value…*]

Sets an attribute to a value for a prefix.

```
nrcmd> dhcp-prefix example-pref set address=ff00::/10
```

dhcp-prefix *name* **unset** *attribute*

Unsets an attribute to a value for a prefix.

dhcp-prefix *name* **get** *attribute*

Gets the explicit value of an attribute for a prefix.

dhcp-prefix *name* **show**

Shows the values of all attributes assigned to a prefix.

dhcp-prefix list

Lists all refixes and any attributes assigned to them.

dhcp-prefix listnames

Lists the names of all prefixes.

dhcp-prefix *name* **listLeases**

Lists the leases associated with the specified prefix name.

dhcp-prefix *name* **addReservation** *ip6address*[*/prefix-length*] *duid*

Adds a lease reservation to the specified prefix name.

dhcp-prefix *name* **removeReservation** {*ip6address*[*/prefix-length*] | *duid*}

Removes a lease reservation from the specified prefix name.

dhcp-prefix *name* **listReservations**

Lists all lease reservations associated with the specified prefix name.

Attributes

Table 2-17 describes the **dhcp-prefix** command attributes.

Table 2-17 dhcp-prefix Command Attributes

Attribute	Usage	Description
address	create set get	Prefix address. Required, no default.
dhcp-type	set get unset	Type of prefix with respect to DHCP. The values are static, dhcp, or prefix-delegation. Optional, default dhcp.
embedded-policy	set get unset	Policy embedded in a single prefix. Optional, no default.
expiration-time	set get unset	The time at which the prefix expires. The server does not allow any new leases to be granted or existing leases to be renewed with a valid lifetime beyond this time. Once the expiration-time has passed, the prefix is no longer used (although old leases and leases with grace or affinity periods continue to exist until those periods elapse). Optional, no default.

Table 2-17 dhcp-prefix Command Attributes (continued)

Attribute	Usage	Description
ignore-declines	enable **disable** get	Controls whether the DHCP server processes a DHCPv6 DECLINE message referencing an IPv6 address or delegated prefix on this prefix. If this attribute is enabled, the DHCP server ignores all declines for leases in this prefix. If this attribute is 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. Optional, default disabled, so that DECLINE messages are processed normally.
link	set get unset	Link associated with the prefix. Optional, no default.
name	create set get	Prefix name to assign. Required, no default.
policy	set get unset	Reference to a shared policy to use when replying to clients. Optional, no default.
prefer-interface-identifier	enable **disable** get	If true, non-temporary addresses are generated for clients using their interface identifier, unless that address is assigned to a different client. If unavailable, a random address is generated. If false, non-temporary addresses are randomly generated for clients. Optional, default disabled.
range	set get unset	Subrange from which prefixes, used for DHCP address assignment, can be configured to assign addresses. Optional, no default.
selection-tags	set get unset	Comma-separated list of selection tags associated with the prefix. Optional, no default.
vpn-id	set get unset	ID of the VPN containing the prefix. Optional, no default.

Related Commands dhcp, dhcp-link

dhcp-prefix-policy

Use the **dhcp-prefix-policy** command to edit a DHCP policy that is embedded in a dhcp-prefix. An embedded policy is a collection of DHCP option values and settings associated with another object, in this case a dhcp-prefix. A dhcp-prefix-policy is created implicitly when you first reference it, and is deleted when the address-block is deleted.

dhcp-prefix-policy *name* **delete**

dhcp-prefix-policy *name* **set** *attribute=value* [*attribute=value ...*]

dhcp-prefix-policy *name* **get** *attribute*

dhcp-prefix-policy *name* **disable** *attribute*

dhcp-prefix-policy *name* **enable** *attribute*

dhcp-prefix-policy *name* **show**

dhcp-prefix-policy *name* **setLeaseTime** *time-value*

dhcp-prefix-policy *name* **getLeaseTime**

dhcp-prefix-policy *name* **setOption** {*opt-name* | *id*} *value*

dhcp-prefix-policy *name* **getOption** {*opt-name* | *id*}

dhcp-prefix-policy *name* **unsetOption** {*opt-name* | *id*}

dhcp-prefix-policy *name* **listOptions**

dhcp-prefix-policy *name* **setVendorOption** {*opt-name* | *id*}*opt-set-name value*

dhcp-prefix-policy *name* **getVendorOption** *name* **setVendorOption** {*opt-name* | *id*} *opt-set-name*

dhcp-prefix-policy *name* un**setVendorOption** {*opt-name* | *id*} *opt-set-name*

dhcp-prefix-policy *name* **listVendorOptions**

Attributes

Table 2-18 describes the dhcp-prefix-policy command attributes.

Table 2-18 dhcp-prefix-policy Command Attributes

Attribute	Usage	Description
affinity-period	set get unset	For DHCPv6, specifies for how long a lease that has become available is retained for a client before it is deleted. This allows a client to obtain an expired lease if the client returns during this period; or it can prevent a client from reusing an address if it returns within this period (if either inhibit-all-renews or inhibit-renews-at-reboot is enabled). Optional, no default.

Table 2-18 dhcp-prefix-policy Command Attributes (continued)

Attribute	Usage	Description
allow-client-a-record-update	enable **disable**	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. Optional, default is FALSE.
allow-client-hints		If allow-client-hints is true, addresses and prefixes requested by the client, in SOLICIT and REQUEST messages, are used if possible. If allow-client-hints is false, addresses and prefixes requested by the client are ignored. Optional, the default is false.
allow-dual-zone-dns-update	enable **disable**	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. Optional, the default is FALSE.
allow-lease-time-override	enable **disable**	Indicates that clients may request a specific lease-time. The server will not honor those requested lease-times if this attribute is set to false. The server will not honor a client's lease-time if that time is longer than the server's normal lease-time. Optional, default is disabled.
allow-non-temporary-addresses	**enable** disable	Determines whether DHCPv6 clients can request non-temporary addresses. Optional, default is TRUE.
allow-rapid-commit	enable **disable**	Determines whether DHCPv6 clients can use a 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. This attribute needs special handling in processing the policies. The server checks all prefix policies (both embedded and named) for the link to which the client has access: • 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. Optional, default is FALSE.
allow-temporary-addresses	**enable** disable	Determines wether DHCPv6 clients can request temporary addresses.
default-prefix-length	set get unset	For delegation, specifies the default length of the delegated prefix if it is not explicitly requested by the requesting router (client). The default length must always be less than or equal to the prefix length of the prefix range. Optional, default is 64.
forward-dnsupdate	set get unset	Specifies the forward zone DNS update. Optional, no default.
forward-zone-name	set get unset	Names an optional forward zone to update. Optional, no default.

Table 2-18 dhcp-prefix-policy Command Attributes (continued)

Attribute	Usage	Description
giaddr-as-server-id	enable **disable**	Enables the DHCP server to set the server-id option on a DHCPOFFER and a DHCPACK to the giaddr of the incoming packet, instead of the IP address of the server (as it will by default). This cases 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 may not actually be an address to which the DHCP client can unicast a packet. In these cases, the DHCP client cannot renew a lease, and must always performing a rebind operation (where the DHCP client broadcasts a request instead of unicasting it to what it believes is the DHCP server). This feature is disabled by default.
grace-period	set get unset	Defines the length of time between the expiration of a lease and the time it is made available for re-assignment. Optional, default is 5m.
inhibit-all-renews	enable **disable**	Permits clients to renew their leases, but the server will force them to obtain new addresses each time they reboot. Optional, default is FALSE.
inhibit-renews-at-reboot	enable **disable**	Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot. Optional, default is FALSE.
limitation-count	set get unset	Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active leases. Optional, no default.
longest-prefix-length	set get unset	For delegation, the longest allowable length for prefixes. If the requesting router (client) requests a prefix length that is longer than this, the value set in this attribute is used instead. Optional, the default is the default-prefix-length.
offer-timeout	set get unset	Tells the server to wait the specified amount of time if it has offered a lease to a client but the offer is not accepted. At the end of the specified time interval, the server makes the lease available again. Optional, default is 2m.
packet-file-name	set get unset	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. Optional, no default.
packet-server-name	set get unset	Identifies the host-name of a server to used 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. Optional, no default.
packet-siaddr	set get unset	Identifies the IP address of the next server in a client's 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. Optional, no default.
permanent-leases	enable **disable**	Indicates that leases for this scope should be permanently granted to requesting clients. Optional, default is disabled.

Table 2-18 dhcp-prefix-policy Command Attributes (continued)

Attribute	Usage	Description
preferred-lifetime	set get unset	Specifies the default and maximum preferred lifetime for leases to DHCPv6 clients. Optional, default value is 1w.
reverse-dnsupdate	set get unset	Specifies the reverse zone DNS update. Optional, no default.
server-lease-time	set get unset	Tells the server for how long a lease is valid. For more frequent communication with the client, it may be useful to have the server consider leases leased for a longer period than the client does. 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. Optional, no default.
shortest-prefix-length	set get unset	For delegation, the shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, the value set in this attribute is used instead. Optional, the default is the default-prefix-length.
split-lease-times	enable **disable**	If enabled, the DHCP server uses the value of the *server-lease-time* attribute internally. Clients are still offered lease times that reflect the configured lease-time option from the appropriate policy, but the server bases its decisions about expiration on the *server-lease-time* value. Optional, default is disabled.
unavailable-timeout	set get unset	Permits the server to make a lease unavailable for the specified period of time and then to return the lease available state. Optional. 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	enable **disable**	When checking the server's database for IP addresses that are reserved, the server by default uses the MAC address of the DHCP client as the key for the database lookup. If *use-client-id-for-reservations* is enabled, then the check for reserved leases is performed by using the client-id of the DHCP client. The client-id is usually supplied by the DHCP client. In cases where it was not supplied by the DHCP client, then it is synthesized by the server, and that value will be used. Optional, the default is disabled.
v4-bootp-reply-options	set get unset	Lists the options that are returned to all BOOTP clients, whether or not a client specifically asks for the option data. Optional, no default.
v4-reply-options	set get unset	Lists the options that are returned to all DHCPv4 clients, whether or not a client specifically asks for the option data. Optional, no default.
v6-reply-options	set get unset	A list of options that should be returned in any replies to DHCPv6 clients.
valid-lifetime	set get unset	Specifies the default and maximum valid lifetime for leases to DHCPv6 clients. Optional, default value is 2w.

Usage Guidelines

You can set individual option values with the **setOption** command, unset option values with the **unsetOption** command, and view option values with the **getOption** and **listOptions** commands. When you set an option value the DHCP server will replace any existing value or create a new one as needed for the given option name.

Related Commands

policy,**client-policy**,**client-class-policy**, **dhcp-address-block-policy**, **dhcp-link-policy**, **scope-policy**, **scope-template-policy**

dhcp-subnet

Use the **dhcp-subnet** command to view and manipulate the current DHCP subnets, which the server creates with the **dhcp-address-block** command. All **dhcp-subnet** command actions take effect immediately. The *subnet-number* value includes the IP address and mask.

dhcp-subnet *subnet-number* **force-available**

dhcp-subnet *subnet-number* **get** *attribute*

dhcp-subnet *subnet-number* [**show**]

Syntax Description

See Table 2-19 on page 2-73 for the **dhcp-subnet** command attributes and their descriptions.

dhcp-subnet *subnet-number* **force-available**

Makes a currently held DHCP subnet available, even a subnet marked as unavailable. Because using the **force-available** action may compromise the integrity of your IP address allocations, ensure that before you use this command, the client assigned the subnet is no longer using it.

dhcp-subnet *subnet-number* **get** *attribute*

Gets the explicit value of an attribute for a DHCP subnet. See Table 2-29 on page 2-114.

dhcp-subnet *subnet-number* [**show**]

Shows the DHCP subnet attributes for a specific address.

Attributes

Table 2-19 describes the **dhcp-subnet** command attributes and their values. They are all read-only attributes.

Table 2-19 dhcp-subnet Command Attributes

Attribute	Usage	Description
address	get	DHCP subnet's address (including the mask). Required, no default.
all-vpns	get	This attribute configures failover for the subnet. If true, this subnet is configured to use the same failover configuration for all VPNs.
client-domain-name	get	Domain name that the client specifies in its messages (if any).
client-flags	get	Either *client-valid* or *client-id-created-from-mac-address* (the client ID was created for internal use from the client's MAC address).
client-host-name	get	Hostname that the client specifies (if any).
client-id	get	Client ID of the DHCP subnet's client.
client-last-transaction-time	get	Time that the client most recently contacted the DHCP server.
client-mac-addr	get	MAC address that the client presents to the DHCP server.
expiration	get	Expiration time of the DHCP subnet's binding.
high-water	get	Highest utilization level recorded since the last statistics.
in-use-addresses	get	Number of addresses that the client currently uses.
last-transaction-time	get	Time at which the client last communicated with the server about the DHCP subnet.

Table 2-19 dhcp-subnet Command Attributes (continued)

Attribute	Usage	Description
relay-agent-option	get	Contents of the DHCP *relay-agent-info* option from the most recent client interaction.
selection-tags	get	String that the client presented when it last leased or renewed the DHCP subnet binding.
state	get	DHCP subnet's state, which can be none=0, available=1, other-available=2, offered=3, leased=4, expired=5, released=6, unavailable=7, or pending-available=8.
unusable-addresses	get	Number of addresses marked unusable.
vpn-id	get	ID of the VPN that contains the DHCP subnet.

Related Commands **acl**, **dhcp-address-block**

dns

The **dns** command sets and enables or disables DNS server attributes. Note that in Network Registrar there is only one DNS server per cluster, hence you do not need to reference the server by name.

dns enable *attribute*

dns disable *attribute*

dns set *attribute*=*value* [*attribute*=*value*…]

dns unset *attribute*

dns get *attribute*

dns [**show**]

dns addRootHint *name ipaddress* [*ipaddress*…]

dns removeRootHint *name*

dns listRootHints

dns addException *name ipaddress* [*ipaddress*…]

dns removeException *name*

dns listExceptions

dns addForwarder *ipaddress* [*ipaddress*…]

dns removeForwarder *ipaddress*

dns listForwarders

dns flushCache

dns removeCachedRR *owner* [*type* [*data*]]

dns rebuildRR-Indexes

dns forceXfer {**primary** | **secondary**}

dns scavenge

dns getStats [**all** | {[**performance**] [**query**] [**security**] [**errors**] [**maxcounters**]} [**sample**]]

dns serverLogs show

dns serverLogs nlogs=*value* **logsize=***value*

Note See the "server" section on page 2-168 for other server commands, including logging.

Syntax Description

See Table 2-20 on page 2-78 for a list of the **dns** command attribute descriptions.

dns enable *attribute*

Enables a DNS server attribute.

dns disable *attribute*

Disables a DNS server attribute, such as to disable NOTIFY for all the zones.

```
nrcmd> dns disable notify
```

dns set *attribute=value* [*attribute=value*…]

Sets one or more attributes of the DNS server.

dns unset *attribute*

Unsets the value of a DNS attribute.

dns get *attribute*

Gets the explicit value of an attribute for the DNS server.

dns [**show**]

Shows all the DNS server attributes.

dns addRootHint *name ipaddress* [*ipaddress*…]

Adds the named root server at a specific IP address using the root hint method. After you specify these servers, Network Registrar queries them for their root NS records that resolve other names. These values need not be exact, but should be accurate enough for the DNS server to retrieve the correct information.

```
nrcmd> dns addRootHint a.root-servers.net 192.168.0.4
```

dns removeRootHint *name*

Removes a root nameserver.

```
nrcmd> dns removeRootHint a.root-servers.net
```

dns listRootHints

Lists a root nameserver.

dns addException *name ipaddress* [*ipaddress*…]

Adds an exception server at a specific IP address.

```
nrcmd> dns addException blue.com. 192.168.1.4
```

dns removeException *name*

Removes an exception server.

dns listExceptions

Lists all the exception nameservers.

dns addForwarder *ipaddress* [*ipaddress*…]

Adds the IP address of any nameservers that you want your Network Registrar DNS server to use as forwarders. Network Registrar forwards recursive queries to these servers before forwarding queries to the Internet-at-large. Note that you can use the exception method to override forwarding for specific domains.

```
nrcmd> dns addForwarder 192.168.1.4
```

dns removeForwarder *ipaddress*

Removes a forwarder server at the IP address.

dns listForwarders

Lists all the forwarder servers.

dns flushCache

Flushes the cache file to stop it from growing. The behavior depends on whether your DNS server is running or stopped.

dns removeCachedRR *owner* [*type* [*data*]]

Removes non-authoritative resource records from in-memory and persistent (non-authoritative) cache:

- With the *type* omitted, removes the entire name set.
- With the *type* included without *data*, removes the resource record set.
- With both *type* and *data* included, purges the specific resource record.

See the attributes in the **addRR** syntax description.

dns rebuildRR-Indexes

Rebuilds indexes for DNS resource records.

dns forceXfer {primary | secondary}

Forces full zone transfers for every zone with the type specified in the command (primary or secondary), regardless of the SOA serial numbers, to synchronize the DNS data store. If a normal zone transfer is already in progress, the command schedules a full zone transfer for that zone immediately after the normal zone transfer finishes.

dns scavenge

Causes scavenging on all zones that have the *scvg-enabled* attribute enabled.

dns getStats [**all** | {[**performance**] [**query**] [**security**] [**errors**] [**maxcounters**]} [**sample**]]

Displays the DNS server statistics generated by the total counters since the last server restart. You can request four categories of statistics, with one qualifying keyword:

- **all**—Displays all statistics available for all DNS servers. Cannot be used with any other category.
- **performance**—Displays performance statistics available for the DNS server. Can be combined with the other categories.
- **query**—Displays query statistics available for the DNS server. Can be combined with the other categories.
- **security**—Displays security statistics for the DNS server. Can be combined with the other categories.
- **errors**—Displays error statistics for the DNS server. Can be combined with the other categories.
- **maxcounters**—Displays the maximum counter statistics for the DNS server. Can be combined with the other categories.
- **sample**—If this keyword is used with one or more categories, displays the last snapshot taken of the counter values.

dns serverLogs show

Shows the number of log files configured and the maximum size of each.

dns serverLogs nlogs=*value* **logsize=***value*

Sets the two server logging parameters: the number of log files (nlogs) and the maximum size of each (logsize). When you use this command, you must specify one or both of the attributes. When you set the logsize, append a K to the value to signify thousands of units or append an M to signify millions of units. For example:

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

Attributes

Table 2-20 describes the **dns** command attributes and their values and defaults, if any.

Table 2-20 dns Command Attributes

Attributes	Usage	Description
activity-counter-interval	set get unset	The sample time interval that server activity counters use in collecting metrics. Optional, default 5m.
activity-counter-log-settings	set get unset	Logs DNS server activity counters in different categories. Optional, default categories are *total*, *performance*, *query*, *errors*, *security*.
activity-summary-interval	set get unset	Time (in seconds) between activity summary log messages, if enabled in the *activity-summary* setting in *log-settings*. Optional, default 5m.
auth-db-cache-kbytes	set get unset	Size of internal cache that the authzone database uses. (Servers use the authzone database to look up answers for zone for which it is authoritative.) The value is rounded up to the nearest 4KB boundary, and a value below 36 is rounded up to 36 KB. Optional, default 5120KB.
axfr-multirec-default	enable **disable** get	Default multirecord full zone transfer (AXFR) choice for remote servers not found in the remote server list. Optional, default disable.
cache-db-cache-kbytes	set get unset	Size of internal cache that the cache database uses. (Servers use the cache database to persist the answers to queries already obtained.) The value is rounded up to the nearest 4-KB boundary, and a value below 36 is rounded up to 36 KB. Applies only if *persist-mem-cache* is enabled. Optional, default 5120 KB.
checkpoint-interval	set get unset	Interval (in seconds) at which to checkpoint zones (take the latest snapshot in the zone checkpoint database). The checkpoint interval set at the zone level overrides this value. Required, default 3h.
collect-sample-counters	**enable** disable get	Switch to turn on or off counter sampling. Optional, default enabled.
default-negcache-ttl	set get unset	Time (in seconds) that negative answers are cached if there is no SOA resource record in the authority section of the reply. An SOA record in the authority section of a negative answer overrides this attribute value (see IETF RFC 2308). Optional, default 0.

Table 2-20 dns Command Attributes (continued)

Attributes	Usage	Description
delegation-only-domains	set get unset	Limits zones to contain NS resource records for subdomains, but no actual data beyond their own apex (for example, SOA records and apex NS record sets). This filters out "wildcard" or "synthesized" data from authoritative nameservers whose undelegated (in-zone) data is of no interest. Not enforced for answers coming from forwarders. Optional, no default.
fake-ip-name-response	**enable** disable get	Controls whether the server, if queried for a domain name that resembles an IP address (for example, an A record like 192.168.40.40), automatically responds with a NXDOMAIN status without even trying to query (or forward to) other servers. Required, default enable.
forward-retry-time	set get unset	Retry interval for forwarding DNS queries to a forwarder or resolution exception server. These queries are recursive and can require more time for the forwarder to resolve. To ensure trying all forwarders, this value should be set to the *request-expiration-time* divided by one less than the total number of configured forwarders. Optional, default 8s.
ha-dns-comm-timeout	set get unset	Amount of time required to determine that an HA DNS partner became unreachable. A partner was determined unreachable after network communication was not acknowledged during the specified time interval. Optional, default 30s.
ixfr-enable	**enable** disable	Controls the incremental transfer behavior for zones for which you did not configure a specific behavior. If incremental transfer is enabled, then you must also set the value of the *ixfr-expire-interval* attribute or accept the default value. Required, default enable.
ixfr-expire-interval	set get	Longest interval to maintain a secondary zone solely with incremental transfers. After this period, the server requests a full zone transfer. Required, default 1w.
local-port-num	set get	UDP and TCP port number on which the DNS server listens for queries. Required, default port 53.
log-settings	set get	Determines which events to log, as set using a bit mask. See Table 2-21 on page 2-83. Logging additional details about events can help analyze a problem. However, leaving detailed logging enabled for a long period can fill the log files and affect server performance. Required, default is all settings except scavenge-details, tsig-details.
max-cache-ttl	set get	Maximum amount of time to retain cached data. Required, default 1w.
max-dns-packets	set get	Maximum number of packets that the DNS server handles concurrently. Required, default 500.
max-negcache-ttl	set get	Sets an upper bound on the amount of time that a Network Registrar DNS server caches a negative response. (This attribute replaces the *neg-cache-ttl* attribute used in previous versions of Network Registrar.) A value of 0 indicates no upper bound. Required, default 1h.
mem-cache-size	set get	Size of the memory cache, in kilobytes. Required, default 10000 (10MB).

Table 2-20 dns Command Attributes (continued)

Attributes	Usage	Description
neg-cache-ttl	set get	Time, in seconds, to cache data learned from other nameservers about nonexistent names or data. Use with pre-6.0 releases only; replaced by the *max-negcache-ttl* attribute. Required, default 10m.
no-fetch-glue	enable **disable**	Controls whether you want the DNS server, when composing a response to a query, to fetch missing glue records. Glue records are A records with the address of a domain's authoritative nameserver. Normal DNS responses include NS records and their A records related to the name being queried. Required, default disable.
notify	**enable** disable	Controls sending NOTIFY messages for zones incurring a change. You must also set the other *notify-* attributes or accept their defaults. Required, default enable.
notify-defer-cnt	set get	With NOTIFY enabled, the maximum number of UPDATE changes to accumulate during the *notify-wait* period. If this number is exceeded, Network Registrar sends notification before the *notify-wait* period passes. Required, default 100 changes.
notify-min-interval	set get	With NOTIFY enabled, the minimum interval required before sending notification of consecutive changes on the same zone to a particular server. Required, default 2s.
notify-rcv-interval	set get	With NOTIFY enabled for secondary zones, the minimum amount of time between the completion of processing of one notification (serial number testing or zone transfer) and the start of processing of another notification. Required, default 5s.
notify-send-stagger	set get	With NOTIFY enabled, the interval to stagger notification of multiple servers of a particular change. Required, default 1s.
notify-source-address	get set unset	Source IP address from which the DNS server sends notify requests to other servers. A value of 0.0.0.0 indicates that the operating system uses the best local address based on the destination. Optional, no default.
notify-source-port	get set unset	UDP port number from which the DNS server sends notify requests to other servers. A value of zero indicates that a random port should be used. If unset, then queries are sent from the port used to listen for queries (see the *local-port-num* attribute). Optional, no default.
notify-wait	set get	With NOTIFY enabled, the period of time to delay, after an initial zone change, before sending change notification to other nameservers. Use this attribute to accumulate multiple changes. Required, default 5s.
persist-mem-cache	**enable** disable	Controls whether the server writes memory cache to, or reads it from, the persistent cache database. See also the *cache-db-cache-kbytes* attribute. Required, default enable.
query-source-address	set get	Source IP address from which the DNS server sends queries to other servers when resolving names for clients. A value of 0.0.0.0 indicates that the operating system will use the best local address, based on the destination. Required, no default.

Table 2-20 dns Command Attributes (continued)

Attributes	Usage	Description
query-source-port	set get	UDP port number from which the DNS server sends queries to other servers when resolving names for clients. A value of zero indicates the need to choose a random port. If this attribute is unset, the port used to listen for queries sends the queries (see the *local-port-num* attribute). Required, no default.
remote-port-num	set get	UDP and TCP port to which the DNS server sends queries to other servers. Required, default port 53.
request-expiration-time	set get unset	Expiration time of general DNS queries: zone transfer SOA queries, and IXFR and notify requests. Note that this value should be considerably larger than the *request-retry-time* value to allow multiple attempts on multiple servers, using exponential backoff. Optional, default 1m 30s.
request-retry-time	set get unset	Retry time interval for general DNS queries: zone transfer SOA queries, and IXFR and notify requests. Note that this is a minimum retry time; the server applies an exponential backoff on retries. Optional, default 4s.
restrict-query-acl	set get	Limits query clients based on the source IP address, source network address, or access control list (ACL). The ACL can contain another ACL or a TSIG key. This global ACL also serves as a filter for nonauthoritative queries. If a query is targeted at an authoritative zone, the corresponding zone's *restrict-query-acl* applies. However, if the query targets no authoritative zone, the global one applies. Required, default **all** (allow all queries).
restrict-recursion-acl	set get	Access control list (ACL) of IP addresses, network addresses, TSIG keys, or other ACLs, that controls for which DNS clients to honor recursive queries. By default, all clients' recursive queries are honored. If a provided list excludes a client and that client requests a recursive query, the server responds as if the client originally requested an iterative (nonrecursive) query. Having one of the deprecated attributes *no-recurse* or *hide-subzones* enabled overrides this setting. Required, default **any**.
restrict-xfer-acl	set get	Default access control list (ACL) that designates who is allowed to receive zone transfers. The zone value overrides this setting. Required, default none.
round-robin	**enable** disable	Controls whether to round-robin equivalent records in responses to queries. Equivalent records are records of the same name and type. Because clients often only look at the first record of a set, enabling this attribute can help balance loads and keep clients from forever trying to talk to an out-of-service host. Required, default enable.
save-negative-cache-entries	**enable** disable	Controls whether to have the server store negative-query-results cache entries in its cache.db file. If disabled, the server discards negative cache entries evicted from the in-memory cache instead of storing them in the cache.db file. Required, default enable.
scvg-ignore-restart-interval	set get	Interval, in seconds, for which a server restart does not recalculate a start scavenging time. Required, default 2h.

Table 2-20 dns Command Attributes (continued)

Attributes	Usage	Description
scvg-interval	set get	With scavenging enabled, the interval, in seconds, at which the zone is scheduled for scavenging. The zone setting of the same attribute overrides this setting. Required, default 1w.
scvg-no-refresh-interval	set get	With scavenging enabled, the interval, in seconds, during which actions, such as dynamic updates, do not refresh the timestamp on a record. The zone setting of the same attribute overrides this setting. Required, default 1w.
scvg-refresh-interval	set get	With scavenging enabled, the interval, in seconds, during which the record can have a timestamp refreshed. The zone setting of the same attribute overrides this setting. Required, default 1w.
simulate-zone-top-dynupdate	enable **disable**	For Windows 2000 Domain Controller compatibility, when processing a dynamic update packet that attempts to add or remove A records from the name of a zone, responds as if the update was successful, rather than with a refusal, as would normally occur from the static/dynamic name conflict. No update to the records at the zone name actually occurs, although the response indicates that it does. Required, default disable.
slave-forward-retry-time	set get	Pertains to forwarding a DNS query in slave mode. These queries are recursive and can require more time for the forwarder to resolve. To ensure that all forwarders are tried, this value should be set to the *request-expiration-time* divided by one less than the total number of configured forwarders. Note that this attribute has no relevance when *slave-mode* is disabled; if disabled, the *forward-retry-time* is used instead. Required, default 30s.
slave-mode	enable **disable**	Controls whether the server should be a slave server that relies entirely on forwarders for data not in its cache. This attribute has no effect unless you also specify the corresponding forwarders. Note that you can override slave mode for specific domains with the DNS exception method. Required, default disable.
subnet-sorting	enable **disable**	Controls whether to re-order address records in responses to queries based on the subnet of the client. Because clients often only look at the first record of a set, enabling this attribute can help localize network traffic onto a subnet. This attribute applies only to answers to queries from clients located on the same subnet as the DNS server. Required, default disable (as implemented in BIND 4.9.7).
tcp-query-retry-time	set get unset	Retry time for DNS queries over a TCP connection (in response to truncated UDP packets). This value should be less than the *request-expiration-time* value. Optional, default 10s.
transfer-source-address	set get unset	Source IP address from which the DNS server sends transfer and SOA requests to other servers. A value of 0.0.0.0 indicates that the operating system uses the best local address based on the destination. Optional, no default.
transfer-source-port	set get unset	UDP port number from which the DNS server sends transfer and SOA requests to other servers. A value of 0 indicates that a random port should be used. If unset, then queries are sent from the port that listens for queries (see the *local-port-num* attribute). Optional, no default.

Table 2-20 dns Command Attributes (continued)

Attributes	Usage	Description
traps-enabled	set get unset	Determines the traps that this server is configured to emit. Optional, no default.
update-acl	set get	Access control list (ACL) for allowing DNS updates to the server. Deprecated in favor of the **update-policy** command (see the "update-policy" section on page 2-190). Use the **!** symbol for negation, for example: `nrcmd> dns set update-acl=acl1,!acl2` See the "acl" section on page 2-2 for the types of ACLs. Setting the attribute at the zone level overrides the server setting. (This attribute replaces the *dynupdate-set* attribute from the previous release, but has been since replaced by the **update-policy** command.) Optional, default is **none**.
update-relax-zone-name	enable **disable**	Controls relaxing of 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 the exact name of the zone. Required, default disable.
version	get	Gets the current software version of the DNS server. Read-only.

DNS Log Settings

Table 2-21 describes the flags you can set with the *log-settings* attribute All the settings are enabled by default except the *scavenge-details* and *tsig-details* settings. If you make changes to the settings, reload and restart the server.

Table 2-21 DNS Log Flags

Flag	Logs
activity-summary	Server activity at intervals set by the *activity-summary-interval* attribute (default five minutes).
config	Server configuration and de-initialization (unconfiguration).
config-details	Generates detailed information during server configuration.
datastore	Datastore processing that provides insight into various events in the server's embedded databases.
ddns	High level dynamic update messages.
ddns-details	Resource records added or deleted due to DNS updates.
ddns-refreshes	DNS update refreshes for Widows 2000 clients.
ddns-refreshes-details	Resource records refreshed during DNS updates for Windows 2000 clients.
lame-delegation	Lame delegation events, although enabled by default. Disabling this flag could prevent the log from getting filled with frequent lame delegation encounters.
notify	NOTIFY transactions.
query-errors	Errors encountered while processing DNS queries.

Table 2-21 DNS Log Flags (continued)

Flag	Logs
root-query	Queries and responses from root servers.
scavenge	Zones scavenged of dynamic resource records.
scavenge-details	More detailed scavenged zone output (disabled by default).
server-operations	General high server events, such as those pertaining to sockets and interfaces.
tsig	Allows logging of events associated with transaction signature (TSIG) DDNS updates.
tsig-details	Causes more detailed logging pertaining to TSIG to be displayed (disabled by default).
xfr-in	Inbound full and incremental zone transfers.
xfr-out	Outbound full and incremental zone transfers.

Configuring a Caching-Only Server

Configuring a DNS server to be caching-only requires the following settings:

- Increase the in-memory cache to the maximum possible value that is aligned to the operating system's physical memory capacity by setting the *mem-cache-size* value:

  ```
  nrcmd> dns set mem-cache-size=200MB
  ```

- When using a large cache, you might want to disable persisting the cache, so that reload time is not impacting the need to save the large cache. Disable the *persist-mem-cache* attribute:

  ```
  nrcmd> dns disable persist-mem-cache
  ```

- Restrict queries to certain clients only by setting the *restrict-query-acl* attribute:

  ```
  nrcmd> dns set restrict-query-acl=myaccesslist
  ```

Defining Forwarders for the Server

Use the **dns addForwarder** command to specify the address (or space-separated addresses) of nameservers you want your Network Registrar DNS server to use as forwarders:

```
nrcmd> dns addForwarder 192.168.50.101
```

Use the **dns enable slave-mode** command to designate the server as a slave:

```
nrcmd> dns enable slave-mode
```

To list the current forwarders, use the **dns listForwarders** command. To edit your forwarder list, you must delete any offending forwarder and re-enter another one. To remove a forwarder or list of forwarders, use the **dns removeForwarder** command:

```
nrcmd> dns listForwarders
nrcmd> dns removeForwarder 192.168.50.101
```

Adding Root Nameservers

Use the **dns addRootHint** command to add root nameservers by name and address. Do this only if the server was inadvertently removed from the list or if there was an update to the list since the last version:

```
nrcmd> dns addRootHint a.root-servers.net. 192.168.0.4
```

Be careful in removing any root servers from the list. If you accidentally remove the address of one of the roots, or you know that it might have changed, use the **nslookup** tool to find out what it is:

```
nslookup a.root-servers.net
```

You can also use the **dig** tool, if it is installed as part of BIND, to update the root servers list. Finally, you can FTP to the ftp.rs.internic.net site to get the latest roots list:

```
dig @a.root-servers.net . ns
ftp ftp.rs.internic.net
<login>
ls domain
<roots list>
```

Adding and Removing Exception Lists

Use the **dns listExceptions** command to list the available exceptions. Then, use the **dns addException** command to add the exception domains and servers, separated by spaces. Use this command only if you do not want your DNS server to use the standard name resolution for names outside the local authoritative zone:

```
nrcmd> dns listExceptions
nrcmd> dns addException blue.example.com. 192.168.60.1 192.168.70.1
```

To remove a resolution exception, use the **dns removeException** command. To replace it, follow this with a **dns addException** command with the new values. You must also flush the cache so that the server does not refer to the old resolution values in cache:

```
nrcmd> dns removeException blue.com.
nrcmd> dns addException blue.com. 192.168.1.8 192.168.1.9
nrcmd> dns flushCache
```

Enabling and Fine Tuning Incremental Zone Transfers

Use the **dns enable ixfr-enable** command to enable incremental transfer for all zones for which you did not configure specific behavior. By default, the *ixfr-enable* attribute is enabled:

```
nrcmd> dns enable ixfr-enable
```

Use these commands to fine tune IXFR:

- **zone** *name* **disable ixfr**—Disables incremental transfer for a single secondary zone if you do not want to use the global value from the **dns disable ixfr-enable** command, unless you override it:

  ```
  nrcmd> zone boston.example.com. disable ixfr
  ```

- **remote-dns** *ipaddr* **create** and **disable ixfr**—Prevents the specified server from performing incremental zone transfers:

  ```
  nrcmd> remote-dns 192.168.1.15 create
  nrcmd> remote-dns 192.168.1.15 disable ixfr
  ```

- **dns set ixfr-expire-interval**—Defines the interval, in seconds, in which to attempt incremental zone transfers, followed by full zone transfers:

  ```
  nrcmd> dns set ixfr-expire-interval=7000
  ```

Related Commands server, zone

dns-interface

The **dns-interface** command adds, removes, and lists Network Registrar DNS interfaces. A DNS interface is a logical representation of the hardware interface (such as a server's Ethernet or Token Ring network interface card). The DNS server uses the configured address information to determine which interface to use to send and receive packets. A DNS server automatically discovers its interfaces and the list of available addresses on those interfaces.

dns-interface *name* **create** *attribute*=*value* [*attribute*=*value...*]

dns-interface *name* **delete**

dns-interface *name* **enable** *attribute*

dns-interface *name* **disable** *attribute*

dns-interface *name* **set** *attribute*=*value* [*attribute*=*value...*]

dns-interface *name* **unset** *attribute*=*value*

dns-interface *name* **get** *attribute*

dns-interface *name* **show**

dns-interface list

dns-interface listnames

Syntax Description

See Table 2-22 on page 2-87 for the **dns-interface** command attributes and their descriptions.

dns-interface *name* **create** *attribute*=*value ...*

Creates a DNS interface and optionally assigns attribute values. The name and attribute values are required for this command.

dns-interface *name* **delete**

Deletes a DNS interface.

dns-interface *name* **enable** *attribute*

Enables an attribute on the specified DNS interface.

dns-interface *name* **disable** *attribute*

Disables an attribute on the specified DNS interface.

dns-interface *name* **set** *attribute*=*value*

Sets an attribute to a value for a DNS interface.

dns-interface *name* **unset** *attribute*=*value*

Unsets the value of a DNS attribute.

dns-interface *name* **get** *attribute*

Gets the explicit value of an attribute for the specified DNS interface.

dns-interface *name* [**show**]

Shows the values of all attributes assigned to the DNS interface.

dns-interface list

Lists all DNS interfaces and any attributes assigned to them.

dns-interface listnames

Lists the names of all DNS interfaces.

Attributes

Table 2-22 describes the **dns-interface** command attributes and their values and defaults, if any.

Table 2-22 dns-interface Command Attributes

Attribute	Usage	Description
address	set get unset	IP address and subnet mask of the interface or interfaces that the DNS server uses. If you do not assign a value to this attribute, the interface is excluded from matching against the list of automatically discovered interfaces. Optional, no default.
name	create set get	Name of the interface that the DNS server uses.
ip6address	set get unset	IPv6 address and prefix length of the interface or interfaces that the DNS server uses. If you do not assign a value to this attribute, the interface is excluded from matching against the list of auto-discovered interfaces. Optional, no default.
port	set get unset	UDP and TCP port number that the DNS server listens on. Optional, default port 53.

Related Commands

dns

dns-update-map

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.

dns-update-map *name* **create** *dhcp-server dns-server dns-update-config* [*attribute*=*value*...]

dns-update-map *name* **delete**

dns-update-map *name* **set** *attribute*=*value* [*attribute*=*value*...]

dns-update-map *name* **unset** *attribute*=*value*

dns-update-map *name* **get** *attribute*

dns-update-map *name* **show**

dns-update-map list

dns-update-map listnames

Syntax Description

See Table 2-23 on page 2-89 for the **dns-update-map** command attributes and their descriptions.

dns-update-map *name* **create** *dhcp-server dns-server dns-update-config* [*attribute*=*value* ...]

Creates a DNS update map and optionally assigns attribute values.

dns-update-map *name* **delete**

Deletes a DNS update map.

dns-update-map *name* **set** *attribute*=*value*

Sets an attribute to a value for a DNS update map.

dns-update-map *name* **unset** *attribute*=*value*

Unsets the value of a DNS update map.

dns-update-map *name* **get** *attribute*

Gets the explicit value of an attribute for the specified DNS update map.

dns-update-map *name* [**show**]

Shows the values of all attributes assigned to the DNS update map.

dns-update-map list

Lists all DNS update maps and any attributes assigned to them.

dns-update-map listnames

Lists just the names of all DNS update maps and any attributes assigned to them.

Attributes

Table 2-23 describes the **dns-update-map** command attributes and their values and defaults, if any.

Table 2-23 dns-update-map Command Attributes

Attribute	Usage	Description
dhcp-client-class	set get unset	Name of the DHCP client-class. Optional, no default.
dhcp-named-policy	set get unset	If the use-named-policy value is used for the *dhcp-policy-selector* attribute, the name of the DHCP policy. Optional, no default.
dhcp-policy-selector	set get	How to select a policy. The options are use-named-policy (the default), use-client-class-embedded-policy, or use-scope-embedded-policy. Required, default use-named-policy.
dhcp-servers	set get unset	Cluster that includes the DHCP server or DHCP failover server pair. Required, no default.
dns-config	set get	Name of the DNS update configuration (see the **dhcp-dns-update** command). Required, no default.
dns-servers	set get unset	Cluster that includes the DNS server or HA DNS server pair. Required, no default.
dns-update-policy-list	set get unset	Comma-separated list of DNS update policies (see the **update-policy** command). Optional, no default.
dns-update-acl	set get unset	Update ACL to apply to zones referenced by the DNS update configuration (*dns-config*) in this map. If 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* is constructed enabling just the *dhcp-servers* to perform DNS updates, using the IP addresses of the single DHCP server or failover pair, and, if specified, the server-key from the *dns-config* attribute. (See the **dhcp-dns-update** command.) Optional, no default.
name	create set get	Name of the DNS update map. Required at creation, no default.

Related Commands

dhcp-dns-update, **dns**

exit

The **exit** command writes all unsaved changes to the database and then terminates the current **nrcmd** session. If Network Registrar cannot save your changes, it displays an error code. The **quit** command is equivalent to the **exit** command.

exit

quit

Note It is good practice to make the last line of code in a batch file an explicit **exit** command. Also terminate this line in the batch file with an end-of-line character.

Related Commands **quit**, **save**

export

The **export** command exports Network Registrar DHCP and DNS server information.

export addresses file=*CSV-text-file*
[**vpn=***name*]
[**config=***config-file*]
[**dhcp-only**]
[**time-ascii | time-numeric**]

export addresses database=*db-name* **user=***username* **password=***password* [**table=***name*]
[**vpn=***name*]
[**config=***config-file*]
[**dhcp-only**]
[**time-ascii | time-numeric**]

export hostfile [*file*]

export leases {**–client | –server**}
[**–vpn** *name*]
[**–time-ascii | –time-numeric**] *file*

export zone *name* {**static | dynamic | both**} *file*

export zonenames {**forward | reverse | both**} *file*

export key *keyname file*

export keys *file*

export option-set *name file*

Syntax Description

export addresses file=*CSV-text-file* [**vpn**=*name*] [**config**=*config-file*] [**dhcp-only**] [**time-ascii | time-numeric**]

Exports all active IP addresses in a comma-separated value (CSV) text file, if specified. If you omit the file, it writes the output in CSV format to the standard output.

Output of the **export** command can include the VPN specification. The value can be any valid, predefined VPN name, or the reserved words *global* and *all*. *Global* indicates all addresses not in any of the defined VPNs. *All* indicates all VPNs, including the global one. If you omit the VPN, the current one applies, as set by the **session set current-vpn** command. If the current VPN is undefined, the global VPN applies. Network Registrar adds the ID of the VPN at the end of each output line in the export file.

Use these conventions for **export addresses** keywords:

- Configuration file—If it exists, the default configuration file is .nrconfig. To use a configuration file other than the default file, use the **config** keyword to identify your configuration file. If there is an [export-addresses] section in the configuration file, the **export** command uses the clusters that the section specifies instead of the default cluster. If you omit a configuration file, the **export addresses** command looks for a default .nrconfig file. This is the same configuration file that the **report** command uses. Network Registrar looks for the file first in your current directory, then in your home directory, and finally in the *install-path*/conf directory. It uses the first file it encounters.

Each line of the configuration file must begin with the character # (comment), a section header enclosed in square brackets, or a parameter=value pair or its continuation. For example:

```
[export addresses]
clusters=machine1 username password, machine2 username password [...]
```

Network Registrar strips leading white space from each line and ignores blank lines.

- **dhcp-only**—This keyword causes the command to output only DHCP information and not DNS information.
- Database tables—The **table** keyword specifies the database table to which the command exports address information. If you omit this keyword, Network Registrar writes to the default table name ip_addresses. If the table already exists in the specified database when you run the **export** command, Network Registrar clears (and resets the columns) before writing the new data. Network Registrar does not provide a warning or confirmation if it clears an existing table.
- Date and time—The optional **time-ascii** and **time-numeric** keywords specify how to output date/time fields to a CSV text file and when the target database does not support the timestamp data type. The default is **time-ascii**.

export addresses database=*db-name* **user**=*username* **password**=*password* [**table**=*name*] [**vpn**=*name*] [**config**=*config-file*] [**dhcp-only**] [**time-ascii** | **time-numeric**]

Exports all active IP addresses into a database table. See the *Network Registrar User's Guide* for the database output format of the **export addresses** command.

export hostfile [*file*]

Creates a host file, in UNIX host file format, from all the zones in the server, ignoring reverse zones. It creates hostfile records from A records, CNAME records, and HINFO records. Each host file record consists of the IP address, FQDN, aliases created from the A and CNAME records, and comments created from HINFO records.

export leases {**–client** | **–server**} [**–vpn** *name*] [**–time-ascii** | **–time-numeric**] *file*

Writes the state of all the current leases to the output file. The **export leases –client** command exports only leased leases. The **export leases –server** command exports any lease with an associated client.

```
nrcmd> export leases -client leaseout.txt
nrcmd> export leases -server leaseout.txt
```

The **–time-ascii** options writes the lease time as a string in the month, day, time, year format; for example Apr 15 16:35:48 2002. The **–time-numeric** option writes lease times as integers representing the number of seconds since midnight GMT Jan 1, 1970; for example, 903968580.

The file is either the name of the output file or a dash (–) for the standard output for client side exports. You cannot use the dash with the **–server** keyword. In addition, the server side export does not permit any nonalphanumeric character such as a dot (**.**) in filenames. For the syntax of the entries in the output file, see the *Network Registrar User's Guide*.

Note Leases exported with the **–client** and **–server** options can show different results when also using DNS updates, if there is a conflict with the client's host name. This occurs because the lease exported with the **–client** option shows the host name requested by the client, while a lease exported with the **–server** option shows the host name the server uses to perform the DNS update.

export zone *name* {**static** | **dynamic** | **both**} *file*

Writes the specified DNS zone into a file in the BIND zone file format, where *name* is the zone whose data you want to write to a file. This example exports the contents of the example.com zone to the hosts.local file:

```
nrcmd> export zone example.com. static hosts.local
```

export zonenames {**forward** | **reverse** | **both**} *file*

Exports just the zone names for a particular zone type—forward, reverse, or both—to a file.

export key *keyname file*

Exports a single transaction signature (TSIG) key that is configured on the cluster to a file. Generates a key definition in BIND syntax so that it can be imported into other clusters or copied into BIND configurations. See the "key" section on page 2-106.

export keys *file*

Exports all the TSIG keys that are configured on the cluster to a file. Generates key definitions in BIND syntax so that they can be imported into other clusters or copied into BIND configurations.

export option-set *name file*

Exports a named DHCP option set to a file.

Related Commands **import**, **key**, **report**

extension

The **extension** command configures and integrates user-written DHCP extensions into the DHCP server. See the *Network Registrar User's Guide* for details on extensions and extension point programming. The order in which to configure an extension is:

1. Write the extension module in Tcl, C, or C++.
2. Create the extension file in the server scripts directory.
3. Configure the DHCP server to recognize the extension, using the **extension** command.
4. Attach the extension to one or more extension points, using the **dhcp attachExtension** command.

extension *name* **create** *language file entry* [**init-args=***value* **init-entry=***value*]

extension *name* **delete**

extension *name* **set** *attribute***=***value* [*attribute***=***value*…]

extension *name* **unset** *attribute*

extension *name* **get** *attribute*

extension *name* [**show**]

extension list

extension listnames

Note The DHCP server reads extensions only when you reload the server. So if you change an extension, you must reload the DHCP server.

Syntax Description

See Table 2-24 on page 2-95 for the **extension** command attribute descriptions.

extension *name* **create** *lang file entry* [**init-args**=*value* **init-entry**=*value*]

Creates a client and optionally assigns initial entry point attributes. The command line attributes are:

- *lang*—Language in which the extension or module is implemented, either Tcl or Dex. Required, no default.
- *file*—Filename relative to the directory extensions in the installation, as an absolute pathname, but cannot contain a sequence of two dots (**..**). Required, no default.
- *entry*—Name of the entry point for the module. This function is called from any extension point to which this module is bound. The arguments for this function are server-implementation-specific. Required, no default.
- For the initial entry point attributes, see Table 2-24 on page 2-95.

This example configures an extension named ext1 using the Tcl file tclfile1.tcl having the entry mytclentry:

```
nrcmd> extension ext1 create Tcl tclfile1.tcl mytclentry
```

extension *name* **delete**

Deletes an extension.

extension *name* **set** *attribute*=*value* [*attribute*=*value*…]

Sets one or more attributes for the extension.

extension *name* **unset** *attribute*

Unsets the value of an extension attribute.

extension *name* **get** *attribute*

Gets the explicit value of an attribute for the extension.

extension *name* [**show**]

Shows the values of all attributes assigned to the extension identified by *name*.

extension list

Lists all extensions and any attributes assigned to them.

extension listnames

Lists just the extension names.

Attributes

Table 2-24 describes the **extension** command attributes and their values and defaults, if any.

Table 2-24 extension Command Attributes

Attribute	Usage	Description
entry	set get	Name of the entry point for the module. This function is called from any extension point to which this module is bound. Required, no default.
file	set get unset	Filename relative to the directory extensions in the installation, or as an absolute pathname, but cannot contain a sequence of two dots (**..**). Required, no default.
init-args	set get unset	Arguments to pass to the *init-entry* function. Optional, no default.
init-entry	set get unset	Name of the *init-entry* point. If you set it, Network Registrar calls this function when the server loads the module and when the server shuts down. Optional, no default.
lang	set get	Language in which the extension or module is implemented: • **Tcl**—Module is a Tcl extension (tcl7.5) • **Dex**—Module is a shared object with C calling interfaces Required, no default.

Related Commands

dhcp

failover-pair

The **failover-pair** command configures and manages the relationship between a pair of DHCP failover servers.

failover-pair create *main-server-address backup-server-address* [*attribute*=*value*...]

failover-pair delete

failover-pair set *attribute*=*value* [*attribute*=*value*...]

failover-pair get *attribute*

failover-pair unset *attribute*

failover-pair show

failover-pair list

failover-pair name addMatch {*vpn/*} {*address/mask*}

failover-pair name removeMatches {*vpn/*} {subnet/mask}

failover-pair name listMatches

failover-pair listnames

failover-pair sync {**update** | **complete** | **exact**} *attribute*=*value* [*attribute*=*value*...]

Attributes

Table 2-25 describes the **failover-pair** command attributes and their values and defaults, if any.

Table 2-25 failover-pair Command Attributes

Attribute	Usage	Description
backup	set get unset	Cluster that contains the backup server for this failover pair. Optional, no default.
backup-pct	set get unset	Percentage of available addresses that the main server should send to the backup server. You must define this value on the main server. If you define it on a backup server, the value is ignored to enable copying of configurations. The value you define here is used as the default value on a scope, unless you have explicitly configured another value. Optional, default 10%.
backup-server	set get	Primary IP address for the backup server. The distinguished value 0.0.0.0 refers to the local DHCP server. The failover protocol uses explicit IP addresses, and the main and backup server must agree on what IP addresses are used to define a failover pair. You therefore should provide an explicit IP address rather than have the server pick from one of many addresses associated with the server's interfaces. Required for creation, no default.

Table 2-25 failover-pair Command Attributes (continued)

Attribute	Usage	Description
dynamic-bootp-backup-pct	set get unset	If dynamic BOOTP is enabled on scopes, this attribute specifies the percentage of available addresses that the main server should send to the backup server. You must define this value on the main server. If you define it on a backup server, the value is ignored to enable copying of configurations. If you do not define a value, or you set the value to 0 (zero), the system uses the backup-pct value instead. This attribute is distinct from backup-pct. If dynamic BOOTP is enabled on a scope, a server never, even in PARTNER-DOWN state, grants leases on addresses that are available to the other server. The server cannot assume such leases are available again. You cannot use the MCLT attribute with dynamic BOOTP leases. Optional, no default.
failover	**enable** disable get	If this attribute is enabled, the specified failover object is configured for failover. Disabling this attribute disables failover with the attached subnets without changing the fundamental configuration. Optional, default enabled.
load-balancing-backup-pct	set get unset	Percentage of clients to which a backup server responds while in normal communication with the main server. If you set this to greater than 0, this enables failover load balancing, as defined in RFC 3074. You must define this value on the main server. If you define it on a backup server, the value is ignored to enable copying of configurations. Optional, default 0%, which disables failover load balancing.
main	set get unset	Cluster that contains the main server for this failover pair. Optional, no default.
main-server	set get unset	Primary IP address for the main server. The distinguished value 0.0.0.0 refers to the local DHCP server. The failover protocol uses explicit IP addresses, and the main and backup server must agree on what IP addresses are used to define a failover pair. You therefore should provide an explicit IP address rather than have the server pick from one of many addresses associated with the server's interfaces. Required, no default.
mclt	set get unset	Maximum client lead time (MCLT), in seconds. The MCLT determines how much ahead of the backup server the client's lease expiration can be. You must define this value on both the main and backup servers, and the value must be identical on both servers. Optional, default 60 minutes.
persist-lease-data-on-partner-ack	**enable** disable get	If this attribute is set to true, the main server updates the Network Registrar database whenever the backup server provides new lease information. Setting this attribute to false improves the performance of the main server; however, when you restart the main server, it no longer has lease information from the backup server. Thus, the main server may offer all clients one lease period with a renewal time based on the current time plus the MCLT. Optional, default enabled.
poll-lease-hist-interval	set get unset	Period of time between occurrences of polling for lease history. Setting this attribute to 0 disables polling for lease history. Optional, no default.

Table 2-25 failover-pair Command Attributes (continued)

Attribute	Usage	Description
poll-lease-hist-offset	set get unset	Fixed time when polling occurs. For example, setting the offset to 1h (for 1:00 A.M.) and the interval to 2h means that polling occurs every two hours, including 1:00 A.M. each day. To represent midnight, use 0. Optional, no default.
poll-lease-hist-retry	set get unset	If polling fails, the number of times to retry. Optional, no default.
poll-lease-hist-server-first	set get unset	First server to poll for lease history data, either main server or backup server. Optional, default mainserver.
poll-subnet-util-interval	set get unset	Period of time between occurrences of polling for subnet utilization data. Setting this value to 0 (zero) disables polling for subnet utilization data. Optional, no default.
poll-subnet-util-offset	set get unset	Fixed time when polling for subnet utilization data occurs. For example, setting the offset to 1h (for 1:00 A.M.) and the interval to 2h means that polling occurs every two hours, including 1:00 A.M. each day. To represent midnight, use 0. Optional, no default.
poll-subnet-util-retry	set get unset	If polling fails, the number of times to retry. Optional, no default.
poll-subnet-util-server-first	set get unset	First server (mainserver or backupserver) to poll for subnet utilization data. Optional, default mainserver.
safe-period	set get unset	If you set this attribute, you specify the length of time (in seconds) that is safe for both servers in the failover pair to continue issuing leases when the network communication between the two has failed. This enables one of the two servers to enter the PARTNER-DOWN state without requiring an operator command. You must define this value on the main server. If you define it on a backup server, the value is ignored to enable copying of configurations. To set the *safe-period* attribute, you must enable the *use-safe-period* attribute. Optional, no default.
scope-template	set get unset	Scope template associated with the failover pair. Optional, no default.
use-safe-period	set get unset	Enabling this attribute allows one server in the failover pair to automatically enter the PARTNER-DOWN state when network communication between the two servers has failed. You must define this value on the main server. If you define it on a backup server, the value is ignored to enable copying of configurations. Because of the risk of issuing duplicate addresses, use-safe-period is disabled by default.

Usage Guidelines

To use the failover-pair command, you must explicitly specify the interfaces that the main and backup servers use to communicate with each other. You do this by specifying the IP address for these interfaces in the main-server and backup-server attributes. Failover must be configured on the same interfaces that the server is using to service client requests. Two additional attributes, main and backup, enable you to identify the cluster to which each server belongs. You must set these attributes to run the **sync** command. Also check that the clusters are configured to enable synchronization with each other.

Related Commands

dhcp

group

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.

group *name* **create** [**description=***value*]

group *name* **delete**

group *name* **set description=***value*

group *name* **get description**

group *name* **unset description**

group *name* [**show**]

group list

group listnames

group enable

group disable

Attributes

The **group** command supports two attributes:

- name—The name assigned to the group. Required, no default.
- description—A description for this group. Optional.

Related Commands

admin, **role**

ha-dns-pair

Use the **ha-dns-pair** command to define and manage the High Availability relationship between a main and backup DNS server. The *ha-dns-main-server* and *ha-dns-backup-server* attributes specify the IP addresses the servers will use to communicate with each other. Two additional properties, main and backup, must be set to the servers' cluster references to use the sync command. The referenced clusters must also be configured with appropriate connection credentials for the sync command to be successful.

Note that 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), attributes set on the ha-dns-pair will always replace values present on the DNS server object in Complete or Exact mode. Administrator groups are used to associate an admin to one or more roles that control access to operations and data.

ha-dns-pair create *main-cluster/address backup-cluster/address* [*attribute*=*value*...]

ha-dns-pair delete

ha-dns-pair enable *attribute*

ha-dns-pair disable *attribute*

ha-dns-pair set *attribute*=*value*

ha-dns-pair get *attribute*

ha-dns-pair unset *attribute*

ha-dns-pair [**show**]

ha-dns-pair list

ha-dns-pair listnames

ha-dns-pair sync {**update** | **complete** | **exact**} {**from-regional** | **main-to-backup** | **backup-to-main**}

Attributes

Table 2-26 describes the **ha-dns-pair** command attributes.

Table 2-26 ha-dns-pair Command Attributes

Attribute	Usage	Description
backup	set get unset	Cluster reference for the backup server in this DNS HA pair relationship. Optional, no default.
ha-dns	enable **disable** get	Enables or disables HA DNS on the DNS server. Optional, default disabled.
ha-dns-backup-server	create set get	IP address of the backup server. Required at creation, no default.

Table 2-26 ha-dns-pair Command Attributes

Attribute	Usage	Description
ha-dns-main-server	create set get	IP address of the main server. Required at creation, no default.
main	set get unset	Cluster reference for the main server in this DNS HA pair relationship. Optional, no default.
name	create set get	Name of the DNS HA pair relationship. Required at creation, no default.
simulate-zone-top-dynupdate	enable **disable** get	For Windows 2000 Domain Controller compatibility, when processing a dynamic update packet which tries to add or remove A records from the name of a zone, responds as if the update succeeded, rather than responding with a refusal, as would normally occur due to the static/dynamic name conflict. No update to the records at the zone name actually occurs, although the response indicates that it has. Optional, default disabled.
update-relax-zone-name	enable **disable** get	Enables or disables relaxation of the RFC 2136 restriction on the zone name record in dynamic updates. This feature allows updates to specify a zone name that is any name in an authoritative zone rather than exactly the name of a zone. Optional, default disabled.

Related Commands **dns**, **dns-interface**

help

The **help** command displays the **nrcmd** program online help.

help

help *command* [*section*…]

Tip You can set the screen buffer size and window size to view the entire content of the help item.

Syntax Description

help

If you enter the **help** command without any arguments, Network Registrar displays a list of all the commands.

help *command* [*section*…]

If you specify the **help** command with a command name, Network Registrar displays the help page for the command of that name. Optionally, you can use the *section* attribute to limit the response to a specified section of the command message.

The section names are:

- **synopsis**—Valid syntax for the command
- **description**—Textual description of the command behavior
- **examples**—Examples of the command usage
- **properties**—Descriptions of the command properties (attributes)
- **status**—Description of the status codes that this command returns

This example prints the synopsis section of the help file for the **help** command:

```
nrcmd> help help synopsis
100 Ok
SYNOPSIS
help
help <cmd> [<section> ...]
```

import

The **import** command imports DHCP lease data or a BIND named.boot file into the DNS server.

import keys *file*

import leases *file*

import named.boot *file*

import named.conf *file*

import option-set *file*

Note Use UNIX style pathnames even when running the **import** command on Windows.

If successful, the **import** command prints `100 Ok` both before and after Network Registrar imports the file. The first `100 Ok` means that the command is being processed (without rejection because of existing locks, licensing problems, or command syntax errors). The second `100 Ok` indicates that the command successfully completed its processing.

Before you can import leases, you must:

1. Configure scopes in the DHCP server for the leases to be imported, using the **scope** command.
2. If the host names for the leases are to be dynamically entered in DNS as part of the import, configure zones in the DNS server to allow DNS updates, using the **zone** *name* **enable dynamic** command.
3. Set the DHCP server to import mode so that it does not respond to other lease requests during the lease importing, using the **dhcp enable import-mode** command.
4. After the leases are imported, take the DHCP server out of import mode so that it will respond to other lease requests, using the **dhcp disable import-mode** command.

The DHCP server might not accept a lease, or a communication failure might drop the lease packet. In the latter case, the server retries the import several times and after about a minute, reports a failure. If the import fails, check the DHCP server log file to find the lease that caused the error. Then go back to the import file, delete all lease entries up to and including the offending one, and repeat the lease import.

Syntax Description

import keys *file*

Imports the keys file. The key import generates as many keys as it finds in the import file. For details on keys, see the "key" section on page 2-106.

import leases *file*

Imports the leases in the file to the DHCP server. The client is given the lesser of the lease times:

- In the import file, or
- That the client would receive if it were to acquire a lease using the existing configuration of the DHCP server.

For example, it is 2:00 P.M. and your scope is configured for a one hour lease. According to the file that you import, the lease time does not expire until 5:00 P.M. After you import the file, the lease expires at 3:00 and not at 5:00.

If your import file specifies a DNS zone name, the server does not use the zone name when it updates DNS. If the file specifies a host name, the server uses the host name when updating DNS, unless the host name was overridden by a *host-name* specification in a client or client-class entry.

The only way to indicate to the DHCP server that the client's host name should be in a zone other than the default associated with the scope is to specify that zone in a client or client-class entry.

You can specify the VPN for imported leases at the end of each lease entry in the import file. The VPN must be predefined. See the "owner" section on page 2-129. All leases without explicit VPN entries are assigned to the current (or global) VPN:

```
nrcmd> import leases LeaseIn
```

import named.boot *file*

Imports the BIND 4.*x.x* named.boot file. This points the server to its database files, such as the /etc/named.boot file on UNIX or Windows:

```
nrcmd> import named.boot /etc/named.boot
```

import named.conf *file*

Imports the BIND 8 or BIND 9 named.conf file. This points the server to its database files, such as the /etc/named.conf file on UNIX or Windows:

```
nrcmd> import named.conf /etc/named.conf
```

import option-set *file*

Imports a DHCP option set from a file. (You can also export option sets using the export option-set command.) For example, to import an option set for Preboot Execution Environment (PXE) clients, modify and import a sample file located in the /examples/dhcp directory:

```
nrcmd> import option-set /examples/dhcp/OptionSetPXE.txt
```

Related Commands **dhcp**, **export**, **scope**, **zone**

key

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.

Note The CLI does not propagate keys to both DHCP servers in a failover configuration. You must use the failover tool in the Web UI to do this.

key *name* **create** *secret* [*attribute*=*value* ...]

key *name* **delete**

key *name* **set** *attribute*=*value*

key *name* **unset** *attribute*

key *name* **get** *attribute*

key *name* **show**

key list

key listnames

Syntax Description

See Table 2-27 on page 2-107 for a description of **key** command attributes.

key *name* **create** *secret* [*attribute*=*value* ...]

Creates a TSIG key by associating a key name with a shared secret value. RFC 2845 recommends that the name concatenate, in FQDN format, the names of the hosts using the key. Enter the shared secret value as base64-encoded. You can use the **cnr_keygen** utility to generate key secrets in this form (see the *Network Registrar User's Guide* for details). The key creation format is:

```
nrcmd> key host-a.host-b.example.com. create
       "xGVCsFZ0/6e0N97HGF50eg==" algorithm=hmac-md5 type=tsig time-skew=300s
```

key *name* **delete**

Deletes the specified TSIG key.

key *name* **set** *attribute*=*value* ...

Sets the value of an attribute for the specified TSIG key.

key *name* **unset** *attribute*

Unsets an attribute for the specified TSIG key.

key *name* **get** *attribute*

Gets the explicit value of an attribute for the specified TSIG key.

key *name* **show**

Displays the attributes of the specified TSIG key.

key list

Lists all security keys and their attributes.

key listnames

Lists only the names of the TSIG keys.

Attributes

Table 2-27 describes the **key** command attributes.

Table 2-27 key Command Attributes

Attribute	Usage	Description
algorithm	set get	Encryption algorithm used to sign messages. RFC 2845 and Network Registrar support HMAC-MD5 encryption only. Required, default hmac-md5.
id	set get	Provides an integer identifier for transaction signature (TSIG) keys
secret	set get	Shared secret used to generate message signatures. The longer the secret, the more secure the encryption. Required, no default.
security-type	set get	Type of security used. RFC 2845 and Network Registrar support TSIG only. Required, default TSIG.
time-skew	set get unset	Number of seconds of error permitted in the signature time. This is the amount of time (+ or –) that the time values can differ between when you add the TSIG record on the client and when the server receives it. Optional, default 300s (5m). **Note** Ensure that the clocks between two servers fall within the time skew period.

Related Commands

acl, **dhcp**, **dns**, **export**, **scope**, **zone**

ldap

The **ldap** command associates remote Lightweight Directory Access Protocol (LDAP) servers with Network Registrar and sets their attributes.

ldap *server* **create** *hostname* [*attribute*=*value*…]

ldap *server* **delete**

ldap *server* **enable** *attribute*

ldap *server* **disable** *attribute*

ldap *server* **set** *attribute*=*value* [*attribute*=*value*…]

ldap *server* **unset** *attribute*

ldap *server* **get** *attribute*

ldap *server* **setEntry** *dictionaryattribute-key*=*value*

ldap *server* **unsetEntry** *dictionaryattribute-key*

ldap *server* **getEntry** *dictionaryattribute-key*=*value*

ldap *server* [**show**]

ldap list

ldap listnames

See the *Configuring Clients and Client-Classes* chapter of the *Network Registrar User's Guide* for usage guidelines.

Syntax Description

See Table 2-28 on page 2-110 for the **ldap** command attributes and their descriptions.

ldap *server* **create** *hostname* [*attribute*=*value*…]

Creates a name entry for the LDAP server at the *hostname* (and optionally assigns values to its attributes). This example creates the LDAP server object myserver with a host name of myserver.mycompany.com:

```
nrcmd> ldap myserver create myserver.mycompany.com
```

ldap *server* **delete**

Deletes the entry for an LDAP server:

```
nrcmd> ldap myserver delete
```

ldap *server* **enable** *attribute*

Enables an LDAP server attribute. After you enable an attribute, you can set its values.

ldap *server* **disable** *attribute*

Disables an LDAP attribute.

ldap *server* **set** *attribute*=*value* [*attribute*=*value*…]

Sets one or more attributes for the LDAP server.

ldap *server* **unset** *attribute*

Unsets the value of an LDAP attribute.

ldap *server* **get** *attribute*

Displays the value of an attribute of the LDAP server.

ldap *server* **setEntry** *dictionary*=*value*

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 string keys to string values. The *dictionary* values are:

create-dictionary—Maps LDAP attributes to DHCP lease attributes. If an entry does not exist, this sets entries in this dictionary to the value of its corresponding DHCP lease attribute. Optional, no default.

create-string-dictionary—Maps LDAP attributes to user specified strings. If an entry does not exist, this sets entries in this dictionary to the matching string. Optional, no default.

env-dictionary—The server can retrieve additional LDAP attributes along with client attributes. If any of these are in a query's results, their values are made available to scripts through the request's environment dictionary. This keys the LDAP value by the value in the query env-dictionary. Optional, no default.

query-dictionary—Mapping between the names of LDAP attributes and DHCP attributes. The server tries to retrieve all the LDAP attributes specified in the dictionary. When a query succeeds, the server sets the values for any LDAP attributes that it returns in the corresponding client attribute. Optional, no default.

This attribute also controls the mapping of an LDAP attribute name to the embedded policy. The LDAP attribute name associated with the *embedded-policy* value is used to create an embedded policy. If the server finds the particular LDAP attribute name, it decodes the attribute data as if it were an encoding of the client-embedded policy. For details about LDAP configuration, see the *Network Registrar User's Guide*.

update-dictionary—Maps LDAP attributes to DHCP lease attributes. When an LDAP object is modified, each LDAP attribute present in this dictionary is set to the value of its corresponding DHCP lease attribute. Optional, no default.

ldap *server* **unsetEntry** *dictionary-attribute-key*

Unsets the value of a dictionary attribute. See the **setEntry** syntax description.

ldap *server* **getEntry** *dictionary-attribute-key*=*value*

Retrieves information from various dictionaries in the LDAP server configuration. See the **setEntry** syntax description.

ldap *server* [**show**]

Shows the values of the attributes of the named LDAP server.

ldap list

Lists the names of the remote LDAP servers and any attributes assigned to them.

ldap listnames

Lists just the names of the remote LDAP servers.

Attributes

Table 2-28 describes the **ldap** command attributes and their values and defaults, if any.

Table 2-28 ldap Command Attributes

Attribute	Usage	Description
can-create	enable **disable** unset	Controls whether an LDAP server can create new entries to store lease state updates. Optional, default disable.
can-query	enable **disable** unset	Controls whether to use an LDAP server for client queries. Optional, default disable.
can-update	enable **disable** unset	Controls whether to use an LDAP server to store lease state updates. Optional, default disable.
connections	set get unset	The Network Registrar LDAP facility is multithreaded. Each LDAP object has a configurable number of connections associated with it. 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 (by enabling the *limit-requests* attribute and setting a *max-requests* attribute value). The *connections* attribute is primarily a performance tuning parameter. In some cases, more than one connection can improve overall throughput. The amount depends on the load on the LDAP server. With many applications using LDAP, five connections would be appropriate; with just Network Registrar using LDAP, 25 would be appropriate. Optional, default one connection.
create-object-classes	set get unset	With the *can-create* attribute enabled, Network Registrar names of the object classes inherited by a newly created entry in the directory. Optional, no default.
default-attribute-value	set get unset	String assigned to any LDAP attributes, listed in the create or update dictionaries, that do not have matching lease attributes. You can list these LDAP attributes in the create update dictionaries. If you omit a value, Network Registrar uses the string *default*. Optional, default is *default*.
dn-attribute	set get unset	If the server can construct the distinguished name (DN) of the LDAP object to update (or create) from one of the lease attributes, it formats the specified *dn-attribute* using the *dn-format* string to construct the object filter that specifies the LDAP server to modify. Optional, no default.
dn-create-format	set get unset	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 attributes, the server formats the specified *dn-attribute* using the *dn-format* string. Optional, no default.
dn-format	set get unset	Formats the *dn-attribute* for entry modification. 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 updated from one of the lease's attributes, the server formats the specified *dn-attribute* using the *dn-format* string to construct the query filter. Optional, no default.
hostname	set get unset	Host name of the LDAP server. Required for creation, no default.

Table 2-28 ldap Command Attributes (continued)

Attribute	Usage	Description
limit-requests	**enable** disable unset	Controls whether to limit the number of outstanding queries on each LDAP client connection. Optional, default enable.
max-referrals	set get unset	Limits the number of LDAP referrals the server follows when querying. A value of zero (the default) means "do not follow referrals." Optional, default 0 referrals.
max-requests	set get unset	With **ldap enable limit-requests**, limits the number of outstanding queries of a single LDAP connection. You can improve performance (and avoid swamping the LDAP server) by limiting the number of outstanding queries. For example, if the LDAP server can handle only 100 requests, setting *max-requests*=20 with *connections*=5 might be appropriate. Adjust the parameters one at a time and monitor the results. Optional, default 20.
password	set get unset	Password of a user with access to the parts of the directory that DHCP uses. (You can configure LDAP servers to allow anonymous access, so this is optional.) Optional, no default.
port	set get unset	Port on the remote LDAP server. Optional, no default.
preference	set get unset	Preferential order of LDAP servers, specified as a positive integer. 1 is the highest preference value. Optional, default 1.
query-timeout	set get unset	Number of seconds the DHCP server should wait for a response to individual LDAP query requests. After a query request times out, the DHCP server drops the request and does not process it again on another LDAP connection or server. Note that the *timeout* attribute configures the timeout for LDAP update and create requests. Optional, default 3s.
referral-attr	set get unset	Name of the LDAP attribute that may indicate that an LDAP response is a referral. Optional, no default. The referral may or may not contain the DN for which to query: • If the DN is present (the current server assumes this), it is used as the search path along with a wildcard search scope in the query that follows the referral. • If the DN is not present, the search path is built by formatting the data in the referral attribute with the referral filter, and the existing search scope is used.
referral-filter	set get unset	If the *referral-attr* attribute does not contain a DN, the referral attribute's data is formatted with this filter expression to build a search path, and the existing search scope for the LDAP server is used. Optional, no default.
search-filter	set get unset	Search 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. An optional **%** at the entry level is replaced by the value of the *dn-attribute*. Optional, no default.

Table 2-28 ldap Command Attributes (continued)

Attribute	Usage	Description
search-path	set get unset	Name of an object in the directory to use as a query's starting point. Together, the *search-path* and *search-scope* attributes control the portion of the directory that the server searches. An optional **%** at the entry level is replaced by the value of the *dn-attribute*. Optional, no default.
search-scope	set get unset	Scope of the search. Optional, default subtree. Can be one these values: • **subtree**—Server searches all the children of the search-path (default). • **onelevel**—Server only searches the immediate children of the base object. • **base**—Server only searches the base object itself.
threadwaittime	set get unset	If there are outstanding queries or updates, the interval (in milliseconds) at which each LDAP connection polls for results and processes queries, updates and creates. Optional, default 100 ms.
timeout	set get unset	Seconds that the DHCP server should wait for a response to an individual query. After a query times out, the server may retry another LDAP server connection or drop the query if there is no other connection. Note that timeout values for queries are smaller than those for updates. Optional, default 10s. (Note that a separate *query-timeout* attribute, at a lower visibility level, is set to 3s by default for LDAP query operations.)
update-search-attribute	set get unset	If the DHCP server cannot directly determine the DN of the object to update, it must issue a query to retrieve the DN. In that case, the server uses data in the lease's *search-attribute* attribute and formats it using the *update-search-filter* attribute. Optional, no default.
update-search-filter	set get unset	Formats the *update-search-attribute* attribute. A **%** is required and is replaced with the value of the *dn-attribute*. Optional, no default.
update-search-path	set get unset	Starting point for the portion of the directory that contains the LDAP objects that the server updates. The *update-search-path* and the *update-search-scope* together control the portion of the directory that contains the objects to update. Optional, no default.
update-search-scope	set get unset	The *update-search-path* and the *update-search-scope* together control the portion of the directory that contains the objects to update. Optional, no default. The scope can be: • **subtree**—Server searches all the children of the search-path. • **onelevel**—Server only searches the immediate children of the base object. • **base**—Server only searches the base object itself.
username	set get unset	DN of a user with access to the parts of the directory that DHCP uses. (You can configure LDAP servers to allow anonymous access, so this is optional). Optional, no default.

Related Commands dhcp

lease

Use the **lease** command to view and manipulate the current DHCP leases in the cluster. All **lease** command attributes are read-only, and all actions take effect immediately. The *ipaddress* value can be a simple IP address or can include the VPN, in the syntax *vpnname/ipaddress*. See the "owner" section on page 2-129. You do not need to reload the server with this command.

lease *address* **activate**

lease *address* **deactivate**

lease *address* **send-reservation**

lease *address* **delete-reservation**

lease *address* **force-available**

lease *address* **get-scope-name**

lease *address* **macaddr**

lease *address* **get** *attribute*

lease *address* [**show**]

lease list

lease list –lansegment *address mask*

lease list –macaddr *macaddress*

lease list –subnet *address mask*

Syntax Description

See Table 2-29 on page 2-114 for the **lease** command attribute descriptions.

lease *address* **activate**

Activates a lease, but does not change the state of a lease marked as unavailable. The *ipaddress* value can include the address' VPN, in this slash-separated format:

vpnname/ipaddress

If there is no VPN prefix for the address, the value set by the **session set current-vpn** applies (see the "session" section on page 2-173), or the global VPN if the current VPN is not set:

```
nrcmd> lease 192.168.1.9 activate
```

lease *address* **deactivate**

Prevents a lease from being given out or renewed, but does not change the state of the lease.

lease *address* **send-reservation**

Sends an existing reservation to the server immediately without having to reload the server. Use this keyword in conjunction with the **scope** *name* **addReservation** command.

lease *address* **delete-reservation**

Deletes an existing reservation from the DHCP server immediately without requiring a server reload. To delete the lease from the internal **nrcmd** database, follow this command with the **scope** *name* **removeReservation** command.

lease *address* **force-available**

Makes a currently held lease available, even if the lease marked as unavailable. Because using the *force-available* action may compromise the integrity of your IP address allocations, ensure that, before you use the keyword, the client holding the lease stopped using the lease.

lease *address* **get-scope-name**

Shows the scope to which a lease belongs.

lease *address* **macaddr**

Shows the most recent MAC address associated with a lease. If no MAC address was ever associated with this lease (or if the lease became unavailable), then Network Registrar displays the error message, "302 Not Found."

lease *address* **get** *attribute*

Gets the explicit value of an attribute for a lease.

lease *address* [**show**]

Shows the lease attributes for a specific address.

lease list

Lists all the leases in all VPNs. Note that there is no VPN modifier for this command.

lease list –lansegment *ipaddress mask*

Lists all leases in a LAN segment, including all leases in primary scopes for the address and mask. It also includes all leases in secondary scopes whose primary scope matches the address and mask.

lease list –macaddr *macaddress*

Lists all leases associated with the specified MAC address. Examples of acceptable formats for the MAC address are:

- 1,6,00:d0:ba:d3:bd:3b
- 00:d0:ba:d3:bd:3b
- 00d0bad3bd3b

lease list –subnet *ipaddress mask*

Lists all leases in a DHCP subnet for the network address and mask.

Attributes

Table 2-29 describes the **lease** command attributes and their values. They are all read-only attributes.

Table 2-29 lease Command Attributes

Attribute	Usage	Description
address	get	IP address of the lease, added at creation.
client-binary-client-id	get	Binary form of the client's MAC address, if any.

Table 2-29 lease Command Attributes (continued)

Attribute	Usage	Description
client-dns-name	get	The DHCP server attempted (possibly successfully) to enter this name into the DNS server for this client. It is related to the *client-host-name* attribute, but may not be identical due to name collisions in the DNS server database.
client-domain-name	get	Domain (if any) to which the client's DNS name belongs.
client-flags	get	The *client-flags* attribute value can be any of these flags: • **client-dns-name-up-to-date**—The client DNS name (A record) is current in the DNS server database. • **client-id-created-from-mac-address**—The client-id was created for internal use from the client-supplied MAC address. If this is true, the server does not report it. • **dns-update-pending**—DNS operation is pending for this client. • **in-limitation-list**—The lease is presently in a limitation list using the limitation ID shown. • **reverse-dns-up-to-date**—The reverse (PTR record) DNS entry is current in the DNS database.
client-host-name	get	DNS name that the client requested the DHCP server place into the DNS server.
client-id	get	Client ID that the client specifies, or one that the DHCP server for this client synthesizes (if *client-id-created-from-mac-address* is set in the *client-flags*).
client-last-transaction-time	get	Date and time when the client most recently contacted the DHCP server.
client-mac-addr	get	MAC address that the client presented to the DHCP server.
client-os-type	get	Operating system of the leased 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*. Other examples are: LANMAN Server, LANMAN Workstation, MAC OS, Microsoft Windows, Microsoft Windows 2000 Professional, Microsoft Windows 95, Microsoft Windows 9x, Microsoft Windows for Workgroups, Microsoft Windows NT Advanced Server, Microsoft Windows NT Server, Microsoft Windows NT Workstation 3.51, Microsoft Windows NT Workstation 4.0, Netware, and OS/2.
expiration	get	Date and time when the lease expires.

Table 2-29 lease Command Attributes (continued)

Attribute	Usage	Description
flags	get	Flags for the lease are backup, deactivated, dynamic, or reserved: • **backup**—The state for this lease was recorded by a server whose role was backup with respect to this lease. • **deactivated**—The lease is de-activated, which means that you should not use it. Any client that uses de-activated leases receives a NAK on its next renewal attempt. • **dynamic**—The lease was last written by a server that knew only about the lease because it was created by a send-reservation command. • **reserved**—The lease is reserved for some MAC address. The table that relates MAC addresses to leases is in the scope. Flags can also include *initialized*, *valid*, and *failover-updated*.
lease-renewal-time	get	Minimal time in which the client is expected to issue a lease renewal.
limitation-id	get	This value relates together leases for which there exists a maximum limit on the number of simultaneous active leases allowed. It is defined in the client or client-class.
relay-agent-circuit-id	get	Accesses and manipulates the DHCP *relay-agent* option *circuit-id* suboption data as stored with a response's lease.
relay-agent-option	get	Contents of the DHCP *relay-agent-info* option from the most recent client interaction.
relay-agent-radius-class	get	Contents of the RADIUS class attribute contained in the RADIUS attributes suboption of the DHCP *relay-agent-info* option.
relay-agent-radius-pool-name	get	Contents of the RADIUS framed-pool attribute contained in the RADIUS attributes suboption of the DHCP *relay-agent-info* option.
relay-agent-radius-user	get	Contents of the RADIUS user attribute contained in the RADIUS attributes suboption of the DHCP *relay-agent-info* option.
relay-agent-remote-id	get	Accesses and manipulates the *relay-agent-remote-id* data as stored with a response's lease.
relay-agent-server-id-override	get	IP address in the *server-id-override* suboption of the DHCP *relay-agent-info* option.
relay-agent-subnet-selection	get	IP address in the *subnet-selection* suboption of the DHCP *relay-agent-info* option.
relay-agent-subscriber-id	get	Contents of the *subscriber-id* suboption of the DHCP *relay-agent-info* option.
relay-agent-vpn-id	get	Contents of the *vpn-id* suboption of the DHCP *relay-agent-info* option. For a description of the VPN ID format, see Table 2-51 on page 2-194.
start-time-of-state	get	Date and time when the state last changed to its current value. Use this attribute to determine when the lease was made unavailable.

Table 2-29 lease Command Attributes (continued)

Attribute	Usage	Description
state	get	Current state of the lease. This can be any of these: • **available**—Not currently leased by any client. Any client information is from the most recent client to lease or be offered this lease. • **expired**—The client did not renew the lease and it expired. Upon expiration, the DHCP server schedules the removal of the client's DNS information. • **leased**—Currently leased to the client whose information appears in the lease. • **offered**—Offered to the associated client. In many cases, the database is not written with information concerning offering a lease to a client, because there is no requirement to update stable storage with this information. • **other-available**—Used only when failover is enabled. A lease in this state is available for allocation by the other server in the failover pair, but not available for allocation by this server. • **released**—The client released the lease, but the server was configured to apply a release grace period. The lease is not available until the grace period expires. • **pending-available**—Used only when failover is enabled. A lease in this state is available as soon as this server can synchronize its available state with the other server. • **unavailable**—The lease is unavailable. It was made unavailable because of some conflict. A ping attempt might show that the IP address was already in use by another client, or the DHCP server might notice another DHCP server handing out this lease.
vendor-class-id	get	Client ID specified by the client.
vpn-id	get	Identifier for the VPN, if any.

Related Commands dhcp, lease-notification, owner, scope, session

lease6

Use the **lease6** command to view and manipulate the current DHCP leases in the cluster. All **lease** command attributes are read-only, and all actions take effect immediately. The *ip6address* value is the IPv6 address of the lease, which can also be prefixed with the name of the VPN in which the address resides, if any.

lease6 [*vpn-name*] *ip6address* **activate**

lease6 [*vpn-name*] *ip6address* **deactivate**

lease6 [*vpn-name*] *ip6address* **force-available**

lease6 [*vpn-name*] *ip6address* **get** *attribute*

lease6 [*vpn-name*] *ip6address* [**show**]

lease6 list

Syntax Description

See Table 2-30 on page 2-118 for the **lease** command attribute descriptions.

lease6 [*vpn-name*] *ip6address* **activate**

Activates a lease, but does not change its state if marked as unavailable. You can prefix the *ip6address* value with the VPN in which the address resides, or the keyword **global** to indicate no explicitly defined VPN; otherwise the value set by the **session set current-vpn** applies (see the "session" section on page 2-173).

lease6 [*vpn-name*] *ip6address* **deactivate**

Prevents a lease from being given out or renewed (even in the available state) without changing its state. Making a currently leased lease inactive does not affect its behavior until it expires and becomes available again.

lease6 [*vpn-name*] *ip6address* **force-available**

Forces a lease into the available state. Because using the **force-available** action can compromise the integrity of your IP address allocations, ensure that, before you use the keyword, the client holding the lease stopped using the lease.

lease6 [*vpn-name*] *ip6address* **get** *attribute*

Gets the explicit value of an attribute for a lease.

lease6 [*vpn-name*] *ip6address* [**show**]

Shows the lease attributes for a specific address.

lease6 list

Lists all the DHCPv6 leases in the DHCP server.

Attributes

Table 2-30 describes the **lease6** command attributes and their values. They are all read-only attributes.

Table 2-30 lease6 Command Attributes

Attribute	Usage	Description
binding-end-time	get	Time the lease binding ends.

Table 2-30 lease6 Command Attributes (continued)

Attribute	Usage	Description
binding-iaid	get	Identify Association Identifier (IAID) of the binding.
binding-rebinding-time	get	Earliest time the server requests the client to issue a Rebind request for the binding.
binding-renewal-time	get	Earliest time the server requests the client to issue a Renew request for the binding.
binding-start-time	get	Start time of the binding.
binding-type	get	Type of binding for the lease. The type number matches the DHCPv6 option number.
client-active-leases	get	Number of active leases that the client is currently using.
client-class-name	get	Most recently derived client-class for the client of the lease.
client-id	get	DHCP Unique Identifier (DUID) of the client for the lease.
client-last-transaction-time	get	Date and time when the client most recently contacted the DHCP server.
client-lookup-key	get	Lookup key for the client, either the DUID or the *v6-override-client-id* expression.
client-relay-message	get	Most recently received relayed message, including the complete Relay-Forw messages, if any.
creation-time	get	Time the lease was created.
flags	get	Flags for the lease: • **deactivated**—Lease is deactivated, which means that you should not use it. Any client that uses a deactivated lease is told to stop using it on the next Renew. • **reserved**—Lease is reserved for some DUID. The table that relates DUID addresses to leases is in the prefix. Flags for internal use only are: • **initialized** • **valid** • **not_in_range**
ip6address	get	IPv6 address (or prefix) of the lease, added at creation.
preferred-lifetime	get	Time the address or prefix is no longer preferred.
prefix-name	get	Reference to the prefix containing this lease.
reservation-lookup-key	get	Lookup key that retrieves a reservation for this lease.
start-time-of-state	get	Date and time when the state last changed to its current value. Use this attribute to determine when the lease was made unavailable.

Table 2-30 lease6 Command Attributes (continued)

Attribute	Usage	Description
state	get	Current state of the lease, which can be: • **available**—Client is not currently leasing. • **expired**—Client did not renew the lease, and it expires and will be made available after the grace period expires. • **leased**—Client is currently leasing. • **offered**—Lease is offered to the client. (In many cases, the database is not written with information concerning offering a lease to a client, because there is no requirement to update stable storage with this information.) • **released**— Client 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. • **revoked**—Client can no longer use the lease. • **unavailable**—Lease is unavailable because of some conflict.
state-expiration-time	get	Earliest time the current state is to expire and result in a state transition. Possible state transitions are: • Offered to Deleted (if not reserved) • Leased to Expired • Expired to Available • Released to Available • Available to Deleted (if not reserved)
valid-lifetime	get	Time the address or prefix is no longer valid.
vpn-id	get	ID of the VPN containing this lease.

Related Commands **dhcp**, **lease-notification**, **owner**, **scope**, **session**

lease-notification

Use the **lease-notification** command to receive notification about the number of available addresses in a scope. 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.

lease-notification available={*number* | *percentage***%**}
[**config=***config-file*]
[**leasing-only**]
[**recipients=***recipient*[**,***recipient*] [**mail-host=***name* [**errors-to=***recipient*]]]
[**scopes=**{{*scopename* | *address-range*}[**,***scopename* | *address-range***,** ....]}]
[**vpn=***name*]

Although you can use the **lease-notification** command interactively, its primary use is as an automated command.

Syntax Description

lease-notification available={*number* | *percentage***%**}
[**config**=*config-file*]
[**leasing-only**]
[**recipients**=*recipient*[**,***recipient*] [**mail-host**=*name* [**errors-to**=*recipient*]]]
[**scopes=**{{*scopename* | *address-range*}[**,***scopename* | *address-range***,** ...]}]
[**vpn**=*name*]

Table 2-31 describes the **lease-notification** keywords. Note that keywords and attributes associated with the **recipients** and **scopes** keywords apply only in connection with those keywords. This example specifies scope1 with an available value of 10% and e-mail recipients billy, joe, and jane:

```
nrcmd> lease-notification available=10% scopes=scope1 recipients=billy,joe,jane
       mail-host=mailhost
```

To specify the range of scopes 192.68.1.0 to 192.68.1.255, the configuration file .nrNotification, the recipients administrator, an available value of 13 leases, and the Windows mail host as mailhost, enter:

```
nrcmd> lease-notification scopes=192.68.1.0-192.68.1.255
       config=/home/bob/.nrNotification recipients=admin@comco.com available=13
       mail-host=mailhost
```

Note If successful, the **lease-notification** command prints `100 Ok` both before and after Network Registrar lists the addresses. The first `100 Ok` means that the command is being processed (without rejection because of existing locks, licensing problems, or command syntax errors). The second `100 Ok` indicates that the command successfully completed its processing.

Table 2-31 lease-notification Command Keyword

Keyword	Description
available	Specify 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.

Table 2-31 lease-notification Command Keyword (continued)

Keyword	Description
config	Specify a configuration file. If you omit a configuration file, Network Registrar searches for the default .nrconfig file.
errors-to	If you specify a *mail-host*, you can also specify the e-mail address of the sender of the e-mail to provide a return path for bounced e-mail. The default value is postmaster.
leasing-only	If you specify *leasing-only*, Network Registrar displays only the scopes that can offer leases. If failover is enabled, this includes scopes for which one of these conditions applies: • Role is main and the failover state is NORMAL, COMM-INTERRUPTED, or PARTNER DOWN. • Role is backup and the failover state is COMM-INTERRUPTED or PARTNER DOWN.
mail-host	On Windows, you must specify a mail-host. On Solaris or Linux, the mail host is generally already configured for the **sendmail** program. You can verify that your UNIX system is properly configured by issuing the command **date | mail** *your-email-address* and observing whether or not the date is e-mailed to you. If mail is not configured, you must specify a mail-host.
recipients	If you specify the e-mail addresses of one or more recipients, Network Registrar sends an e-mail report to those addresses. Otherwise, Network Registrar directs the report to standard output.
scopes	Specify the scopes either by their names or as a range or ranges of addresses. Network Registrar checks any scope containing any address that falls within the range of addresses. If you omit any scopes or addresses, Network Registrar checks all scopes that the specified cluster manages.
vpn	If you specify a VPN, you can enter the VPN name or the keywords **all** or **global**. The **all** keyword notifies of addresses in all the configured VPNs. The **global** keyword notifies of all addresses not in any specific VPN.

Related Commands

lease

license

The **license** command adds and distributes license keys across multiple Network Registrar servers. Keys are maintained in a separate license file (product.licenses in the installation's config subdirectory) so that you can easily redistribute keys to other servers by copying the file after the key is correctly entered into the first server. You must enter your license key the first time you configure any cluster:

- Permanent license—You do not see the license message again unless you move your cluster to another machine.
- Evaluation copy of Network Registrar—You have a license that expires.
- Invalid or missing licensing key—You cannot configure or manage the Network Registrar servers. However, the servers themselves continue to function normally.
- License expires in seven or fewer days—You see a warning when you start Network Registrar.

The command syntax is:

license *key* **create**

license *key* **delete**

license *key* **get**

license *key* [**show**]

license list

license listnames

Syntax Description

license *key* **create**

Creates the license. Anyone can set the license key for the first time. Only the Web UI superuser, global administrator, or administrator set up with full access (using the **admin** *name* **set nrcmd-flag=full** command) can reset it. To set a new license key, run the **nrcmd** program in interactive mode, then exit and rerun the **nrcmd** program:

```
$ nrcmd -C cluster1 -N admin -P changeme
nrcmd> license 1234-abcd-5678-efgh create
nrcmd> exit
```

license *key* **delete**

Deletes the license.

license key get

Gets the key values for the license. With limited access (using the **admin** *name* **set nrcmd-flag=limited** command), you cannot set or get the attributes, except initially.

license *key* [**show**]

Shows the encrypted license key value.

license list

Lists the license keys.

license listnames

Lists the names of the license keys.

option

Use the **option** command to configure DHCP option definitions. Use the reserved names **dhcp-config** and **dhcp6-config** to view the currently configured option sets for DHCPv4 and DHCPv6, respectively. Use the reserved names **dhcp-custom** and **dhcp6-custom** to view, add, modify, or delete custom option definitions. (You can also use **dhcp-config** and **dhcp6-config** to add, modify, and 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 you use the **save** command.

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.

To view the suboptions of a complex option definition, you must use the Web UI.

option *id option-set* **create** *option-name type* [*attribute*=*value*…]

option {*name* | *id*} *option-set* **delete**

option {*name* | *id*} *option-set* **set** *attribute*=*value*…

option {*name* | *id*} *option-set* **unset** *attribute*

option {*name* | *id*} *option-set* **get** *attribute*

option *option-set* **list**

option *option-set* **listnames**

option listtypes

Syntax Description

Table 2-32 describes the **option** command attributes and their values.

option *id option-set* **create** *option-name type* [*attribute*=*value*…]

Creates the option definition, using the option ID, option set to which it belongs, name of the option, type (8-bit or 16-bit), and an optional attribute definition.

option {*name* | *id*} *option-set* **delete**

Deletes the option definition.

option {*name* | *id*} *option-set* **set** *attribute*=*value* [*attribute*=*value*…]

Sets option attributes and their values.

option {*name* | *id*} *option-set* **unset** *attribute*

Unsets an option attribute value.

option {*name* | *id*} *option-set* **get** *attribute*

Gets an option attribute value.

option *option-set* **list**

Lists all the options and their attributes in an option set.

option *option-set* **listnames**

Lists just the names of the options in an option set.

option listtypes

Listing the allowed values for the "base-type" attribute when creating an option definition

Attributes

Table 2-32 describes the **option** command attributes and their values.

Table 2-32 option Command Attributes

Attribute	Usage	Description
base-type	set get unset	Data type of the option, which can be: AT-INT8 AT_SINT8 AT_SHORT AT_SSHORT AT_INT AT_SINT AT_IPADDR AT_STRING AT_NSTRING AT_BOOL AT_DNSNAME AT_IP6ADDR AT_RDNSNAME AT_INTI AT_SINTI AT_SHRTI AT_SSHRTI AT_VENDOR_OPTS AT_VENDOR_CLASS AT_VENDOR_NOLEN AT_ZEROSIZE Required, no default.
default-value	set get unset	Default value for the option, stored in raw form. Should be used when the attribute is unset. If the value has the flag AF-INITIALIZE, this value should be set in newly created objects. Optional, no default.
deprecated	set get unset	If the option is deprecated, the product compatibility version in which it was deprecated. Optional, no default.
desc	set get unset	Description of the option. Optional, no default.
enumerations	set get unset	Optional, no default
feature-id	set get unset	If the option is part of a feature set, the ID of the option. Optional, no default.
flags	set get unset	General set of flags identifying the option. Optional, no default.
id	create set get	ID of the option. Required during creation, no default.

Table 2-32 option Command Attributes (continued)

Attribute	Usage	Description
name	create set get	Name of the option. Required during creation, no default.
optional	enable disable get	Enable if this is an optional option. Optional, no default.
repeat	set get unset	Repeat count for the option, which can be ZERO_OR_MORE, ONE_OR_MORE, or EVEN_NUMBERED. Optional. The default value is 0 (zero) which is logically the same as a value of 1 (one). If you do not set this attribute, the option definition enforces a count of 1.
spec	create set get	Name of the option definition set of which this option is a part. Required at creation, no default.
visibility	set get unset	Visibility of the option. Optional, default 5.

option-set

Use the **option-set** command to create option definition sets.

option-set *name* **create [8-bit | 16-bit] vendor-option-string=***string-value* [*attribute=value*]

option-set *name* **create [8-bit | 16-bit] vendor-option-enterprise-id=***integer* [*attribute=value*]

option-set *name* **delete**

option-set list

option-set listnames

option-set listtypes

option-set *name* **[show]**

option-set *name* **set** *attribute=value ...*

option-set *name* **get** *attribute*

option-set *name* **unset** *attribute*

option-set *name* **enable** *attribute*

option-set *name* **disable** *attribute*

option-set dhcp-custom unset

option-set dhcp6-custom unset

Usage Guidelines

Use the reserved names **dhcp-config** and **dhcp6-config** to view the currently configured option sets for DHCPv4 and DHCPv6 respectively. Use the reserved names **dhcp-custom** and **dhcp6-custom** to view custom option definitions. You cannot use the reserved names to create or delete an option set. You create custom sets when you create the first custom option definition. To clear all custom option definitions, use **unset**.

Changes to custom sets are merged with the built-in option definitions to form the config sets. Modifications to custom sets are not visible in the config sets until you save them.

Use the **listtypes** command to view the list of option types available for use to create custom option definitions.

owner

Use the **owner** command to create, delete, and list the owner of address blocks, subnets, and zones.

owner *tag* **create** *name* [*attribute*=*value*]

owner *tag* **delete**

owner list

owner listnames

owner *tag* **show**

owner *tag* **set** [*attribute*=*value*...]

owner *tag* **get** *attribute*

owner *tag* **unset** *attribute*

owner *tag* **enable** *attribute*

owner *tag* **disable** *attribute*

Attributes

Table 2-33 describes the **owner** command attributes and their values and defaults, if any.

Table 2-33 owner Command Attributes

Attribute	Usage	Description
tag	create set get unset	A unique name for this owner. Typically a short name that represents this owner. Required at creation and has no default value.
name	create set get unset	The full or complete name for this owner. Required at creation and has no default value.
contact	set get unset	Contact information for this owner.
organization	set get unset	The name of the organization as registered with the American Registry of Internet Numbers (ARIN) and used in reporting to ARIN.

policy

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 *Network Registrar User's Guide*.

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. The commands indicated in the syntax as [*embedded*-]**policy** are ones that you can use with or without the hyphenated embedded object type prefix. For named policies, you can create, list, or list names. You cannot use these command elements with embedded policies. For the 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* **create** [*attribute*=*value*…]

policy *name* **create clone=***clone-name*

[*embedded*-]**policy** *name* **delete**

[*embedded*-]**policy** *name* **enable** *attribute*

[*embedded*-]**policy** *name* **disable** *attribute*

[*embedded*-]**policy** *name* **set** *attribute*=*value* [*attribute*=*value*…]

[*embedded*-]**policy** *name* **unset** *attribute*

[*embedded*-]**policy** *name* **get** *attribute*

[*embedded*-]**policy** *name* [**show**]

policy list

policy listnames

[*embedded*-]**policy** *name* **setOption [***opt-name* | *id] value*

[*embedded*-]**policy** *name* **getOption [***opt-name* | *id]*

[*embedded*-]**policy** *name* **unsetOption [***opt-name* | *id]*

[*embedded*-]**policy** *name* **listOptions**

[*embedded*-]**policy** *name* **setV6Option [***opt-name* | *id] value*

[*embedded*-]**policy** *name* **getV6Option [***opt-name* | *id]*

[*embedded*-]**policy** *name* **unsetV6Option [***opt-name* | *id]*

[*embedded*-]**policy** *name* **listV6Options**

[*embedded*-]**policy** *name* **setVendorOption** [*opt-name* | *id] opt-set-name value*

[*embedded*-]**policy** *name* **getVendorOption** [*opt-name* | *id] opt-set-name*

[*embedded*-]**policy** *name* **unsetVendorOption** [*opt-name* | *id] opt-set-name* >

[*embedded*-]**policy** *name* **listVendorOptions**

[*embedded*-]**policy** *name* **setV6VendorOption** [*opt-name* | *id] opt-set-name value*

[*embedded*-]**policy** *name* **getV6VendorOption** [*opt-name* | *id] opt-set-name*

[*embedded*-]**policy** *name* **unsetV6VendorOption** [*opt-name* | *id] opt-set-name*

[*embedded*-]**policy** *name* **listV6VendorOptions** [*vendoroption*]

[*embedded*-]**policy** *name* **setLeaseTime** *value*

[*embedded*-]**policy** *name* **getLeaseTime**

Syntax Description

See Table 2-34 on page 2-133 for the **policy** command attribute descriptions.

Note Be aware that the DHCPv6 options are ignored when set on a policy that affects a prefix.

policy *name* **create** [*attribute*=*value*…]

Creates a policy (and optionally assigns attribute values).

```
nrcmd> policy CompanyB create
```

policy *name* **create clone**=*clone-name*

Creates a new policy based on a copy of the specified policy.

```
nrcmd> policy CompanyB create clone=CompanyC
```

[*embedded*-]**policy** *name* **delete**

Deletes a policy.

[*embedded*-]**policy** *name* **enable** *attribute*

Enables the attribute for a policy.

[*embedded*-]**policy** *name* **disable** *attribute*

Disables the attribute for a policy.

[*embedded*-]**policy** *name* **set** *attribute*=*value* [*attribute*=*value*…]

Sets an attribute to a value for a policy.

```
nrcmd> policy default set grace-period=3d
nrcmd> dhcp-address-block-policy 10.10.0.0/16 set offer-timeout=2m
nrcmd> client-policy 1,6,00:d0:ba:d3:bd:3b set server-lease-time=5d
nrcmd> client-class-policy CableModem set dhcp-reply-options=all-subnets-local
nrcmd> scope-policy testScope set bootp-reply-options=time-offset
nrcmd> dhcp reload
```

[*embedded*-]**policy** *name* **unset** *attribute*

Unsets the value of an attribute for a policy. You cannot unset required attributes.

[*embedded*-]**policy** *name* **get** *attribute*

Gets the explicit value of an attribute for a policy.

[*embedded*-]**policy** *name* [**show**]

Shows the values of all attributes assigned to a policy.

policy list

Lists all policies and any attributes assigned to them.

policy listnames

Lists the names of all policies.

[*embedded*-]**policy** *name* **set**[**V6**]**Option** [*opt-name* | *id] value*

Sets a standard DHCP option name or identifier (*opt-name* or *id)* to a specified value on a policy. When you set an option value, the DHCP server replaces any existing value or creates a new one as needed for the given option name.

```
nrcmd> policy default setOption dhcp-lease-time 608400
```

For a list of all the DHCP options that you can configure, use the **help dhcp-option** command.

[*embedded*-]**policy** *name* **unset**[**V6**]**Option** [*opt-name* | *id*]

Unsets the value of an option for a policy.

[*embedded*-]**policy** *name* **get**[**V6**]**Option**[*opt-name* | *id*]

Gets the explicit value of an option for the policy.

[*embedded*-]**policy** *name* **list**[**V6**]**Options** [*opt-name* | *id*]

Lists the standard options in a policy. (The DHCPv6 options are ignored when set on a policy affecting a prefix.)

```
nrcmd> policy default listOptions
(51)dhcp-lease-time: 604800
```

[*embedded*-]**policy** *name* **set**[**V6**]**VendorOption** [*opt-name* | *id*] *opt-set-name value*

The name of a vendor-specific option definition set

Associates the name of a vendor-supplied DHCP option name or identifier and the vendor supplied option definition set (opt-set-name) with the policy and assigns a value to the option definition set. Use this when a definition is an array that requires braces and square brackets.

[*embedded*-]**policy** *name* **unset**[**V6**]**VendorOption** [*opt-name* | *id*] *opt-set-name*

Deletes the association between the specified policy and a vendor-supplied DHCP option suboption field. Use the suboption-index syntax for arrays.

[*embedded*-]**policy** *name* **get**[**V6**]**VendorOption** [*opt-name* | *id*] *opt-set-name*

Gets vendor-specific option data for the policy.

[*embedded*-]**policy** *name* **list**[**V6**]**VendorOptions**

Lists data for all vendor options in a policy or, optionally, lists data for a specific vendor option.

```
nrcmd> policy 168.1-net listVendorOptions
```

[*embedded*-]**policy** *name* **setLeaseTime** *value*

Sets the client lease time for a policy. The lease time is the value of the *dhcp-lease-time* DHCP option. To view the lease time value, use the [*embedded*-]**policy** *name* **listOptions** command. The time is displayed in seconds.

[*embedded-*]**policy** *name* **getLeaseTime**

Gets the client lease time for a policy.

Attributes

Table 2-34 describes the [*embedded-*]**policy** command attributes.

Table 2-34 policy Command Attributes

Attribute	Usage	Description
affinity-period	set get unset	For DHCPv6, specifies for how long a lease that has become available is retained for a client before it is deleted. This allows a client to retain an expired lease if the client returns during this period; or it can prevent a client from reusing an address if it returns within this period (if either inhibit-all-renews or inhibit-renews-at-reboot is enabled). Optional, no default.
allow-client-a-record-update	enable **disable** unset	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. Optional, default is false.
allow-client-hints	enable disable	If allow-client-hints is true, addresses and prefixes requested by the client, in SOLICIT and REQUEST messages, are used if possible. If allow-client-hints is false, addresses and prefixes requested by the client are ignored. Optional, the default is false.
allow-dual-zone-dns-update	enable **disable** unset	If enabled, the DHCP server returns the *client-fqdn* option (81) so that the client can perform an A record update itself. Also, the server performs an A record update on the client's behalf. This is required to support certain DHCP deployments that represent their clients in two DNS zones. If both the *allow-client-a-record-update* and *allow-dual-dns-update* attributes are enabled, the latter takes precedence. Optional, default disable.
allow-lease-time-override	**enable** disable unset	Clients can request a specific lease time. If this attribute is disabled, the server does not honor requested lease times longer than the server's normal lease time. Optional, default enable.
allow-rapid-commit	set get unset	Allows a DHCPv6 client to use fewer messages to obtain configuration information from its server. Enable this attribute only if a single DHCP server services the client or if unused are not an issue. By default, Rapid Commit is not allowed. If this attribute is set in named or embedded prefix policy, it is ignored.
allow-temporary-addresses	set get unset	Allows the use of standard IPv6 unicast addresses in very short, non-renewable leases. The default is true and allows clients to request temporary addresses.
clone	create	Policy from which the current one is cloned. Can be used during a create operation to create a new policy based on an existing one. **Note** This value is not stored.
default-prefix-length	set get unset	Sets the default length of a delegated prefix when the length is not explicitly stated by the requesting router (client). The value is always less than or equal to the prefix length of the prefix range; that is, a number between 0 and 128. The default is 64.

Table 2-34 policy Command Attributes (continued)

Attribute	Usage	Description
forward-dnsupdate	set get unset	Identifies the forward DNS zone to update.
forward-zone-name	set get unset	Identifies an optional forward zone to update.
giaddr-as-server-id	enable **disable** unset	Causes the DHCP server to set the *server-id* option on a DHCPOFFER and DHCPACK to the *giaddr* of the incoming packet, instead of the address of the server (which is the default). All unicast renews are then sent to the relay agent instead of directly to the DHCP server, and so renews arrive at the DHCP server with option 82 (*relay-agent-info*) data appended to the packet. Some relay agents may not support this capability and in some complex configurations the *giaddr* may not actually be an address to which the DHCP client can unicast a packet. In these cases, the DHCP client cannot renew a lease, and always performs a rebinding operation (where the client broadcasts instead of unicasting a request to what it believes is the DHCP server). Optional, default false.
grace-period	set get unset	Time, in seconds, between the expiration of a lease and the time it is made available for re-assignment. Optional, default 300s (5m).
inhibit-all-renews	enable **disable** unset	Causes the server to reject all renewal requests and forces the client to obtain a new address whenever it contacts the DHCP server. Optional, default disable.
inhibit-renews-at-reboot	enable **disable** unset	Allows clients to renew their leases, but forces them to obtain new addresses each time that they reboot. Optional, default disable.
limitation-count	set get unset	Sets the number of clients with identical limitation keys that are allowed to access your network. Supply an integer value greater than zero (0). The *limitation-id* attribute is set using the **client** command (see the "client" section on page 2-15). Optional.
longest-prefix-length	set get unset	Sets the maximum length of a delegated prefix when the maximum length is not explicitly stated by the requesting router (client). This must always be less than or equal to the prefix length of the prefix range; that is, a number between 0 and 128. The default is the default-prefix-length.
offer-timeout	set get unset	The time, in seconds, that the server waits to re-offer a lease if a client does not accept it. Optional, default 120s (2m).

Table 2-34 policy Command Attributes (continued)

Attribute	Usage	Description
packet-file-name	set get unset	Name of the boot file for a client's boot process. The server returns this filename in the *file* field of its replies. Optional, no default, but cannot be longer than 127 characters. This attribute can also contain these variable substitution values: • *%@docsis-vers%*—If you specify the DOCSIS version value, the server substitutes it with the version presented in the DHCP request packet's vendor-class-identifier option. This version can be either **docsis1.0** or **docsis1.1**. If the vendor-class-id option is missing or does not contain a DOCSIS version string, the server substitutes the docsis-version-id-missing string. See Table 2-9 on page 2-33. • *%@mac-addr%*—If you specify the MAC address value, the server substitutes this string with the source MAC address as presented in the DHCP request packet.
packet-server-name	set get unset	Host name of a server to use in a client's boot process. The server returns this host name in the *sname* field of its replies. Optional, no default, but cannot be longer than 64 characters.
packet-siaddr	set get unset	IP address of the next server in a client's boot process. For example, this might be the address of a TFTP server BOOTP clients use. The server returns this address in the *siaddr* field of its reply. Optional, no default.
permanent-leases	enable **disable** unset	When enabled, grants permanent leases to clients. Optional, default disable.
preferred-lifetime	set get unset	Sets the default and maximum preferred lifetime for leases to DHCPv6 clients. Default is 1w.
reverse-dnsupdate	set get unset	Identifies the reverse DNS zone to update.
server-lease-time	set get unset	Time that the server believes the lease is valid. It may help for the server to consider leasing for a longer period than the client requests so as to get more frequent client communication, along with the stability of long lease times. 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. Optional, no default.
shortest-prefix-length	set get unset	Sets the minimum length of a delegated prefix when the minimum length requested is shorter than this. Explicitly stated by the requesting router (client). This must always be less than or equal to the prefix length of the prefix range; that is, a number between 0 and 128. The default is the default-prefix-length.
split-lease-times	enable **disable** unset	Controls whether the server uses the value of the *server-lease-time* attribute to determine the length of a lease, rather than using the lease time returned to the client. Optional, default disable.
unavailable-timeout	set get unset	Controls the time that a lease remains unavailable before it becomes available again. Optional, default is 1d.

Table 2-34 policy Command Attributes (continued)

Attribute	Usage	Description
v4-bootp-reply-options	set get unset	Lists the options that are returned to all BOOTP clients, whether or not a client specifically asks for the option data. Optional, no default.
v4-reply-options	set get unset	Lists the options that are returned to all DHCPv4 clients, whether or not a client specifically asks for the option data. Optional, no default.
v6-reply-options	set get unset	Sets the list of options that should be returned in any replies to DHCPv6 clients. This attribute will not be used if configured on a named or embedded prefix policy.
valid-lifetime	set get unset	The default and maximum valid lifetime for leases to DHCPv6 clients. Optional, default is 2w.

Related Commands

admin, **client-class**, **client-class-policy**, **client-policy**, **dhcp**, **lease6**, **scope**, **scope-policy**

quit

The **quit** command writes all unsaved changes to the database and then terminates the current **nrcmd** session. If Network Registrar cannot save your changes, it displays an error code. The **exit** command is equivalent to the **quit** command.

quit

exit

Related Commands **exit**, **save**

region

The **region** command configures objects representing a geographic region containing other objects, such as address blocks, subnets, and zones.

region *tag* **create** *name* [*attribute=value*]

region *tag* **delete**

region list

region listnames

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*

Attributes

Table 2-35 describes the **region** command attributes and their values and defaults, if any.

Table 2-35 region Command Attributes

Attribute	Usage	Description
contact	set get unset	Contact information for this region. Optional, no default.
name	create set get	Full or printable name for the region. Required at creation, no default.
scope-template	set get unset	An optional scope template to be used when creating scope objects from subnets associated with this region.

remote-dns

The **remote-dns** command controls the behavior of the DNS server when it is communicating with other DNS servers. Use it either to control incremental zone transfers or send multiple records per TCP packet.

remote-dns *ipaddress*[*/maskbits*] **create** [**ixfr=**{**true** | **false**} | **multirec=**{**true** | **false**}]

remote-dns *ipaddress*[*/maskbits*] **delete**

remote-dns *ipaddress*[*/maskbits*] **enable** {**ixfr** | **multirec**}

remote-dns *ipaddress*[*/maskbits*] **disable** {**ixfr** | **multirec**}

remote-dns *ipaddress*[*/maskbits*] [**show**]

remote-dns list

remote-dns listnames

Syntax Description

remote-dns *ipaddress*[*/maskbits*] **create** [**ixfr=**{**true** | **false**} | **multirec=**{**true** | **false**}]

Creates a remote DNS server description. See the **enable** syntax description for the optional attributes. This example creates the remote server description 192.168.1.1 with the net mask of 255.255.0.0:

```
nrcmd> remote-dns create 192.168.1.1/16
```

Note Each net mask octet is composed of 8 bits. In the previous example, the first two octets are significant, thus the netmask is 16. If the first three octets are significant, the net mask is 24.

remote-dns *ipaddress*[*/maskbits*] **delete**

Deletes a remote DNS server description.

remote-dns *ipaddress*[*/maskbits*] **enable** {**ixfr** | **multirec**}

Enables incremental zone transfers (IXFRs), multiple records, or both for a remote DNS server.

- **ixfr**—Whether a foreign server supports incremental transfer and to query it for incremental (IXFR) before full (AXFR) when asking for zone transfers. Although unwittingly setting this to true is generally harmless, doing so may result in additional transactions to accomplish a zone transfer. Optional, initial default disable.
- **multirec**—Whether to send a remote server zone transfers (AXFR) with multiple records in one TCP packet. Older DNS servers crash when they receive such transfers, despite being allowed by the protocol. Optional, initial default disable.

When you enable or disable incremental transfer, Network Registrar looks for the most specific match. That is, it matches the machine with the longest mask. You can use this attribute to specify a group of servers with a single command.

This example enables all DNS servers on this network to perform incremental transfer:

```
nrcmd> remote-dns create 192.168.0.0/16 ixfr=true
```

This example disables incremental transfers on all DNS servers on this network:

```
nrcmd> remote-dns create 192.168.0.0/16 ixfr=false
```

remote-dns *ipaddress*[/*maskbits*] **disable** {**ixfr** | **multirec**}

Disables incremental zone transfers or multiple records for a remote DNS server. See the **enable** syntax description.

remote-dns *ipaddress*[/*maskbits*] **unset** {**ixfr** | **multirec**}

Unsets the incremental zone transfers or multiple records attribute for the remote DNS server.

remote-dns *ipaddress*[/*maskbits*] [**show**]

Shows the attributes for the remote DNS server.

remote-dns list

Lists all remote DNS server descriptions and any attributes assigned to them.

remote-dns listnames

Lists just the names of the remote DNS servers.

Related Commands **dns**, **server**

report

The **report** command produces a summary of dynamic and static IP address use for one or more clusters.

report [**column-separator=***character-string*] [**dhcp-only**] [**file=***output-file*] [**vpn=***name*]

The output for the **report** command is a table consisting of column-aligned data. There are three types of rows in the table. In all rows, the following information is listed for each scope or subnet:

- Network number (in hexadecimal format).
- Number of high-order bits set in the subnet mask.
- Network number (in canonical dotted-octet format).

For each scope-defined subnet, a row is generated that lists the following information:

- Subnet/Mask.
- Scope name.
- % Free—As a percentage of dynamic addresses available to be leased.
- Total Dynamic—Total number of addresses configured, excluding reservations.
- Total Reserved—Total number of reserved addresses.
- Leased—Number of addresses actively leased by clients.
- Avail—Number of addresses available to be leased.
- Other Avail—Leases set aside for the safe failover partner to lease when communications are interrupted.
- Pending Avail—Leases not available to be leased because the server is in the communications-interrupted failover state.
- In Transition—Leases offered to clients or waiting for the configured grace period before again becoming available.
- Reserved Active—Number of reserved addresses actively leased by clients.
- Unavailable—In a range and marked as unavailable by the server, regardless of flags.
- Active Deactivated—Number of addresses that have been administratively deactivated.

If more than one scope shares a common subnet and mask, the **report** command generates a row that summarizes that subnet. Additionally, for each subnet, the size of which is determined by the default or specified mask-bits, the **report** command creates a row that summarizes any scopes in the subnet, and adds the following information:

- Total—All addresses in the subnet.
- Static—Addresses statically assigned.
- Unallocated—Addresses unallocated to DHCP scope ranges, otherwise reserved or statically assigned, and therefore available for static assignment or allocation to a scope range.

At the end of the report, the **report** command generates a grand total row that summarizes the data from all the subnets.

Syntax Description

report [**column-separator**=*character-string*] [**dhcp-only**] [**file**=*output-file*] [**vpn**=*name*]

When you use the **report** command with no keywords, it creates a report of static DNS addresses and dynamic DHCP addresses for the cluster on which it is running and sends the report to standard output. You can limit the report, redirect it to a file, and change its column delimiters by using the keywords. Table 2-36 describes the **report** command attributes.

Note If successful, the **report** command prints `100 Ok` both before and after Network Registrar lists the addresses. The first `100 Ok` means that the command is being processed (without errors). The second `100 Ok` indicates that the command processing completed successfully.

Table 2-36 report Command Keywords

Keywords	Description
column-separator	Character string that you want the report to use between the columns. The default is a single space. If you specify whitespace, you must precede it with a backslash (\) and, if entering it on the command line, use quotation marks. For example: **"\ "**.
dhcp-only	Summary of just the DHCP information.
file	Filename to which the **report** command writes the output. If you omit a filename, the **report** command appears on the screen. Because it can take a long time to collect DNS data, you should not run the **report** command interactively when requesting DNS summaries.
vpn	VPN address space from which scopes are selected to examine when executing this command. If a <vpn-name> is not specified the current-vpn is used. If the reserved vpn-name ‘global’ is used the global (unnamed) VPN address space is used. Do not use the vpn-name ‘all’ with this command. This is reserved because the report command has no mechanism to distinguish identical IP addresses in different VPNs.

Related Commands

export

role

Use the **role** command to set up and manage administrator roles. A role describes the operations that a group of administrators can perform, and any data constraints that should be applied. You must assign a role to an administrator group to be associated with an administrator.

role *name* **create** *base-role* [*attribute=value*]

role *name* **delete**

role list

role listnames

role *name* **show**

role *name* **set** *attribute=value...*

role *name* **get** *attribute*

role *name* **enable** *attribute*

role *name* **disable** *attribute*

Attributes

Table 2-37 describes the **role** command attributes and their values and defaults, if any.

Table 2-37 role Command Attributes

Attribute	Usage	Description
all-sub-roles	**enable** disable get	If this attribute is unset or enabled, then the value of the *sub-roles* attribute for this role instance is ignored, and the subrole authorization for the role is for all subroles. If disabled, then the *sub-roles* attribute provides the list of subroles for which this role instance is authorized. If the unconstrained attribute is enabled, then the values of this attribute and the values of the *sub-roles* attribute are ignored, and the subrole authorization for the role is for all sub-roles. Optional, default enabled.
constraints	get	List of constraints for the role. Optional, no default. **Note** Use the Network Registrar Web UI to make changes to constraints.
groups	get set	Groups with which this role is associated. Any member of one of these groups can play this role (can perform the operations allowed by this role). Optional, no default.
name	create set get	The name of this role. Required at creation, no default.
read-only	enable **disable**	Enables or disables whether all constraints associated with this role are limited to read-only access. Optional, default disabled.
role	get set	Base role for the role. The base role defines the operations (for example, modifying a zone) that are allowed and the further constraints on these operations (for example, only zones owned by a specific list of owners). Optional, no default.

Table 2-37 role Command Attributes (continued)

Attribute	Usage	Description
sub-roles	get set	List of subroles associated with this role instance. If the *all-sub-roles* attribute is unset or enabled, then this attribute is ignored. If the *all-sub-roles* attribute is disabled, then this attribute specifies the list of subroles for this role instance, and an administrator associated with this role instance has authorization limited to the specified subroles. If the administrator has multiple role instances where the role attribute is the same, then subrole authorization for that role should be taken to be the union of all of the sets of subroles from the individual role instances; and, if any of these role instances has the *all-sub-roles* attribute enabled, then subrole authorization for that role is for all subroles. Also, if any role instance for a matching role has the *unconstrained* attribute enabled, then subrole authorization for that role is for all sub-roles. Optional, no default.
unconstrained	enable **disable** get	Enables or disables whether the role has no other constraints beyond the list of operations it can perform. Optional, default disabled.

Related Commands group, admin

router

The **router** command enables you to configure and manager router attributes.

router *name* **create** *address type* [*attribute*=*value* ...]

router *name* **delete**

router list

router listnames

router *name* **show**

router *name* **get** *attribute*

router *name* **set** [*attribute*=*value* ...]

router *name* **unset** *attribute*

Attributes

Table 2-38 router Command Attributes

Attribute	Usage	Description
address	create set get	The IP address of the management interface for the router you are configuring. Required at creation, no default.
description	create set get	A description of the router you are configuring. Optional, no default.
device-timeout	create set get	Indicates the timeout in seconds. It represents the maximum timeout the RIC server's handler will wait for data from the router. Optional, default is 60.
enable	create set get	The enable password, in clear-text form. See the Caution for the password attribute for information about managing clear-text values. Optional, no default.
enable-secret	create set get	The identifier for the secret containing the clear-text password that enables superuser access to the router.
interfaces	create set get	The list of the interfaces associated with this router. This attribute enables the RIC server to return information about a router and its interfaces, or it enables the CCM server to return a list of routers and interfaces for the UI to display as a tree of routers and interfaces. Optional, no default.
login-template	create set get	The name of a login template that can be used to further customize the RIC server's login and enable interaction sessions. Optional, no default.
login-temp-obj	create set get	The actual login template. You can use this attribute to enable the CCM server to provide the login template to the stateless RIC server. Optional, no default.

Table 2-38 router Command Attributes (continued)

Attribute	Usage	Description
name	create set get	The name of the router you are configuring. Required at creation, no default.
owner	create set get	The owner of the router you are configuring. The owner attribute groups similarly owned routers and can be used to limit administrative access. Optional, no default.
password	create set get	The password used to authenticate the username. Optional, no default. **Caution** In most circumstances, avoid using this attribute. If you must use this attribute, make sure you are using a secure link. The primary use of this attribute is to provide a clear-text password to the RIC server, which intentionally does not have access to the secret storage module.
password-secret	create set get	The identifier for the secure password that encapsulates the clear-text password used to authenticate the username. Optional, no default.
region	create set get	The region associated with this router. The region field is used to group similarly located routers and to limit administrative access. Optional, no default.
type	create set get	The type of router that you are configuring. Use this attribute to enable the RIC server to use the correct implementation of the router-specific interface. Required, no default.
username	create set get	The username used to log in to this router. Optional, no default.
use-ssh	create set get	Determines whether the RIC server must use SSH to communicate with the router. Optional, default is 1 (one).
virtual-router	create set get	This flag indicates that the RIC server does not manage the router object. Use this attribute when there is no network connectivity to the router, or you do not want synchronization with the network configuration. A virtual router configuration is stored in the CCM database, but no attempt is made to push the configuration to the router, or to synchronize any configuration changes from the router. Optional, default is false.

router-interface

The **router-interface** command enables you to configure and manage an interface on a specified router.

router-interface *router name* **create** *router* [*attribute=value ...*]

router-interface *router name* **delete**

router-interface *router* **list**

router-interface *router* **listnames**

router-interface *router name* **show**

router-interface *router name* **get** *attribute*

router-interface *router name* **set** [*attribute=value ...*]

router-interface *router name* **unset** *attribute*

Attributes

Table 2-39 router-interface Command Attributes

Attribute	Usage	Description
bundle-id	set get unset	The identifier of the bundle. This is used for grouping bundled interfaces. Optional, no default.
cable-dhcp-giaddr	set get unset	The setting for giaddr selection in cable interfaces. Optional, default is 0 (zero).
cable-helper	set get unset	The list of IP addresses stored as the cable-helper value on this interface. Optional, no default.
description	set get unset	A descriptive statement about this interface. Optional, no default.
ip-helper	set get unset	The list of IP addresses stored as the ip-helper value on this interface. Optional, no default.
is-master	set get unset	Indicates that this is the master interface in a bundle of interfaces. Optional, default is false.
is-virtual	set get unset	Indicates that this interface is a virtual sub-interface. Optional, default is false.
mac-address	set get unset	The MAC address of this interface. Optional, no default.

Table 2-39 router-interface Command Attributes (continued)

Attribute	Usage	Description
name	create set get unset	The name of the router interface. Optional, no default.
owner	set get unset	The owner of this object. This owner field is used to group similarly owned objects and to limit administrative access. Optional, no default.
parent	set get unset	The parent interface where there are subinterfaces or bundled interfaces configured on the router. Optional, no default.
primary-subnet	set get unset	The primary subnet (and interface address) for this interface. Optional, no default.
region	set get unset	The region associated with this object. This region field is used to group similarly located objects and can be used to limit administrative access. Optional, no default.
router	set get unset	A reference by OID to the router of which this interface is a part. Required, no default.
secondary-subnets	set get unset	The list of secondary subnets (and interface addresses) for this interface. Optional, no default.
state	set get unset	The enabled/disabled state of this interface. Optional, no default.

router-type

Use the **router-type** command to display the available router types.

router-type list

router-type listnames

Attributes

Table 2-40 describes the attributes you can display with the **router-type** command.

Table 2-40 router-type Command Attributes

Attribute	Usage	Description
description	get	An optional description of the specified router type.
manufacturer	get	The name of the manufacturer.
name	get	The name of the specified router type.
router-os-version	get	The version of the operating system that the specified router uses.

save

The **save** command validates and saves your changes to the database.

save

Related Commands server

scope

The **scope** command creates and edits DHCP scopes.

scope *name* **create** *addr mask* [*template=template-name*][*attribute=value…*]

scope *name* **applyTemplate** *template-name*

scope *name* **delete**

scope *name* **enable** *attribute*

scope *name* **disable** *attribute*

scope *name* **set** *attribute=value* [*attribute=value* ...]

scope *name* **unset** *attribute*

scope *name* **get** *attribute*

scope *name* [**show**]

scope list

scope listnames

scope *name* **listLeases**

scope *name* **changeMask** *netmask*

scope *name* **clearUnavailable**

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**

Syntax Description

See Table 2-41 on page 2-153 for the **scope** command attribute descriptions.

scope *name* **create** *ipaddress mask* [*template=template-name*] *attribute=value…*]

Creates a scope (and optionally sets an attribute). Specify the scope mask in base-10 (for example, 255.255.255.0), not in hexadecimal:

```
nrcmd> scope testscope create 192.168.1.0 255.255.255.0
```

scope *name* **applyTemplate** *template-name*

Applies the named scope template to the scope.

scope *name* **delete**

Deletes a scope.

scope *name* **enable** *attribute*

Enables an attribute for a scope.

```
nrcmd> scope testscope enable dynamic-bootp
```

scope *name* **disable** *attribute*

Disables an attribute for a scope.

```
nrcmd> scope testscope disable dynamic-bootp
```

scope *name* **set** *attribute=value* [*attribute=value ...*]

Sets one or more attributes for a scope.

```
nrcmd> scope testscope set ping-timeout=350
```

scope *name* **unset** *attribute*

Unsets an attribute for a scope. You cannot unset required attributes.

scope *name* **get** *attribute*

Gets the explicit value of an attribute for a scope. This example gets the DNS zone name:

```
nrcmd> scope testscope get dns-zone-name
```

scope *name* [**show**]

Shows the values of all attributes assigned to a scope.

scope list

Lists all scopes and any attributes assigned to them.

scope listnames

Lists just the names of scopes.

scope *name* **listLeases**

Lists the leases in a scope. This list can be very long.

scope *name* **changeMask** *netmask*

Changes the network mask of a scope.

```
nrcmd> scope testScope changemask 255.255.254.0
```

scope *name* **clearUnavailable**

Clears the unavailability of leases in a scope to make them all available.

scope *name* **addRange** *start end*

Adds a range of addresses to a scope. The start and end values can be host numbers or IP addresses. Host numbers are relative to the initial address in the DHCP subnet defined by the scope, and full IP addresses must fall within this subnet. If the combined ranges are contiguous, Network Registrar merges them, if possible:

```
nrcmd> scope testScope addRange 192.168.1.10 192.168.1.20
nrcmd> scope testScope addRange 10 20
```

scope *name* **removeRange** *start end*

Removes a range of available addresses in a scope, specified by start and end addresses. If removing a range breaks the address continuity, Network Registrar splits the ranges.

```
nrcmd> scope testscope removeRange 192.168.1.10 192.168.1.15
```

scope *name* **listRanges**

Lists the available addresses in a scope.

scope *name* **addReservation** *ipaddr* {*macaddr* | *lookup-key*} [*-mac*|*-blob*|*-string*]

Adds a reservation to a scope.

```
nrcmd> scope testScope AddReservation 192.168.1.10 1,6,00:d0:ba:d3:bd:3b
```

Tip You can use the **lease** *name* **send-reservation** command to send the reservation immediately to the server without reloading it. For more information, see the "lease" section on page 2-113.

scope *name* **removeReservation** *ipaddr* {*macaddr* | *lookup-key*} [*-mac*|*-blob*|*-string*]

Removes a reservation from a scope, specifying the MAC address or IP address of the client.

```
nrcmd> scope testscope removeReservation 192.168.1.10
```

scope *name* **listReservations**

Lists the reservations in a scope.

Attributes

Table 2-41 describes the **scope** command attributes and their values and defaults, if any.

Table 2-41 scope Command Attributes

Attribute	Usage	Description
addr	get	Address of the DHCP subnet for which this scope contains addresses. Read-only.
allocate-first-available	enable **disable** unset	If enabled, forces all allocation of new IP addresses from the scope to be from the first available address; otherwise (the default), allocation is from the least-recently-used address. If this attribute is not set or is unset, then the DHCP server's *priority-address-allocation* attribute determines whether to allocate the first available address. If *priority-address-allocation* is set in this case, then the scope allocates addresses as if *allocate-first-available* were set. If *allocate-first-available* is enabled or disabled for the scope, then *priority-address-allocation* is ignored. Optional, default disable.

Table 2-41 scope Command Attributes (continued)

Attribute	Usage	Description
allocation-priority	set get unset	Assigns an ordering to scopes, such that IP address allocation takes place from acceptable scopes with a higher priority, until the addresses in all of them are exhausted. Lower values have higher priority (the 0 value, the default, is treated as not having an allocation priority). You can mix scopes with allocation priorities with those without them in the same network, in which case those with an allocation priority are examined first. If *allocation-priority* is not set (or is unset or 0), then the allocation priority of the scope is controlled by the DHCP server's *priority-address-allocation* attribute. If the latter is set in this case, then the allocation priority for the scope is its network number; if not, then scopes are allocated round-robin. If *allocation-priority* is set for the scope, then the server's *priority-address-allocation* is ignored. Optional, default 0 (no allocation priority).
backup-pct	set get unset	Determines the percentage of available addresses that the main server should send 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, this attribute overrides the defined values for 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 percentage is ignored and 50% is used.
bootp	enable **disable** unset	Controls whether the server accepts BOOTP requests. If you want clients to always receive the same addresses, you need to reserve IP addresses for all your BOOTP clients. Optional, default disable.
deactivated	enable disable unset	A scope that does not extend leases to clients. It treats all of the addresses in its ranges as if they were individually de-activated. Optional, no default.
dhcp	**enable** disable unset	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	set get unset	Number of the bytes in a lease's 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. Optional, no default.

Table 2-41 scope Command Attributes (continued)

Attribute	Usage	Description
failover-backup-percentage	set get unset	If the attribute *allocate-first-available* is enabled on a scope and the scope participates in a failover relationship, this value is the address boundary below which the addresses of the failover backup server are allocated. Normal client addresses are allocated in ascending order, while the backup server's addresses are allocated in descending order from this boundary. If unset or zero, then the boundary used for this allocation is halfway between the first and last address configured in the ranges. If there are no available addresses below the boundary, the first one above the boundary is used. Optional.
free-address-config	set get unset	SNMP trap configuration (see the "addr-trap" section on page 2-7). Optional, no default.
ignore-declines	enable **disable** unset	Determines whether the scope should turn off recognition of server lease declines. Optional, default disable.
ping-clients	enable **disable** unset	Controls whether the server should attempt to ping an address. If enabled, also indicate a ping timeout. Optional, default disable.
ping-timeout	set get unset	The number of milliseconds that the DHCP server should wait for ping responses. If you make this value too large, you slow down the lease offering process. If you make this value too small, you reduce the effectiveness of pinging addresses before offering them. Three hundred milliseconds is a frequent compromise. Optional, default 300 ms.
policy	set get unset	Name of the policy associated with the scope. Required, default is the default policy. This means that the scope uses all the properties set in the default policy (including the lease time), unless specifically reset.
primary-subnet	set get unset	Subnet address and mask of the scope's primary scope, used when multiple logical IP subnets are present on the same physical network.
renew-only	enable disable unset	Controls whether to allow existing clients to re-acquire their leases, but not to 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 an available IP address to continue to use it. Optional, no default.
selection-tag-list	set get unset	Comma-separated list of scope-selection tags associated with the scope. The scope compares a client's selection criteria to this list to determine whether the client can obtain a lease from the scope. Replaces the *selection-tags* attribute in Network Registrar 6.2. Optional, no default.
selection-tags	set get unset	Comma-separated list of selection criteria associated with a scope. Deprecated in Network Registrar 6.2 in favor of the *selection-tag-list* attribute. Optional, no default.
subnet	set get	Network address of the DHCP subnet represented by the scope. Required, no default.

Table 2-41 scope Command Attributes (continued)

Attribute	Usage	Description
vpn	set get unset	Virtual attribute instead of the *vpn-id* attribute. When you set this attribute, the ID of the VPN becomes the *vpn-id* attribute value. You can also get the *vpn* of a scope and it returns the name associated with the current *vpn-id*. Optional, no default.
vpn-id	set get unset	ID of the VPN in which the addresses in the scope reside. You must define the VPN using the **vpn** *name* **create** *id* command (see the "owner" section on page 2-129). If unset, the ID of the global VPN is used. Optional, default is the current VPN.

Related Commands **admin**, **client-class**, **client-class-policy**, **client-policy**, **dhcp**, **policy**, **scope-policy**, **scope-selection-tag**

scope-policy

The **scope-policy** command configures DHCP embedded policies for scopes. A scope-policy is a policy object embedded within (and limited to) a scope object. Each scope may contain option data within its embedded policy, and may refer to a named policy with more option data, such as a router IP address.

The DHCP server implicitly creates and deletes embedded scope-policies when you create or delete the corresponding scopes. You manipulate the scope-policy using the name of the corresponding scope.

For the syntax and descriptions, see the "policy" section on page 2-130.

Attributes

See Table 2-34 on page 2-133 for the attribute descriptions.

Related Commands

client-policy, **client-class**, **client-class-policy**, **policy**, **scope**

scope-selection-tag

The **scope-selection-tag** command defines the tags that you add for the scope selection criteria for scopes, clients, and client-classes.

Note Network Registrar no longer requires creation of scope selection tags.

When the DHCP server reads a client entry (from the local database or from LDAP), the server checks its scope-selection inclusion and exclusion criteria against the tags defined for the scopes on this network. If the client entry refers to tags that are not present in any scope in the network, the server handles the tags depending on whether the reference is to include or exclude tags. If the reference is for exclusion, the tags have no effect. If the tags are not present and the reference is for inclusion, the server determines that there is no acceptable scope on that network for this client.

scope-selection-tag *name* **create**

scope-selection-tag *name* **delete**

scope-selection-tag list

Syntax Description

scope-selection-tag *name* **create**

Creates a scope-selection tag.

```
nrcmd> scope-selection-tag internal create
```

scope-selection-tag *name* **delete**

Deletes a scope-selection tag.

```
nrcmd> scope-selection-tag internal delete
```

Note When you delete a tag, Network Registrar removes it from the tag list, but does not remove it from any existing scope, client, or client-class configurations.

scope-selection-tag list

Lists all scope-selection tags.

```
nrcmd> scope-selection-tag list
```

Related Commands

admin, **client-class**, **client-class-policy**, **dhcp**, **scope**

scope-template

Use the **scope-templat**e command to create a template to use when you are setting up multiple scopes.

scope-template *name* **create** [*attribute=value*]

scope-template *name* **create clone=***clone-name*

scope-template *name* **apply-to** {**all** | *scope-name...*}

scope-template *name* **delete**

scope-template *name* **set** [*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**

Syntax Description

See Table 2-42 on page 2-160 for the **scope-template** command attribute descriptions.

scope-template *name* **create** [*attribute=value ...*]

Creates a scope template and optionally assigns attribute values.

scope-template *name* **create clone=***clone-name*

Creates and names a copy of the specified scope template.

scope-template *name* **delete**

Deletes a scope template.

scope-template *name* **apply-to** [**all** | *scope1*,*scope2*,...]

Applies a scope template to one or more scopes.

scope-template *name* **set** *attribute*

Sets one or more attributes for a scope template.

scope-template *name* **get** *attribute*

Gets the explicit value of an attribute for a scope.

scope-template *name* **unset** *attribute*

Unsets the value of the specified attribute.

scope-template *name* **show**

Shows the values associated with a specified scope template.

scope-template *name* **enable** *attribute*

Enables the specified attribute in the named scope template.

scope-template *name* **disable** *attribute*

Disables the specified attribute in the named scope template.

Attributes

Table 2-42 describes the **scope-template** command attributes.

Table 2-42 scope-template Command Attributes

Attribute	Usage	Description
allocate-first-available	set get unset	This boolean attribute forces the 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, then the decision on whether to allocate the first available IP address in the scope is 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 allocates 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	set get unset	Assigns an ordering to scopes, such that IP address allocation takes place from acceptable scopes with a higher priority, until the addresses in all of them are exhausted. Lower values have higher priority (the 0 value, the default, is treated as not having an allocation priority). You can mix scopes with allocation priorities with those without them in the same network, in which case those with an allocation priority are examined first. If *allocation-priority* is not set (or is unset or 0), then the allocation priority of the scope is controlled by the DHCP server's *priority-address-allocation* attribute. If the latter is set in this case, then the allocation priority for the scope is its network number; if not, then scopes are allocated round-robin. If *allocation-priority* is set for the scope, then the server's *priority-address-allocation* is ignored. Optional, default 0 (no allocation priority).
backup-pct	set get unset	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 it is defined on a backup server, it is ignored. This attribute overrides a failover pair's defined values 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 the scope supports *dynamic-bootp*. If you set the value to 0 (zero), 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 failover pair's load-balancing-backup-pct is used.

Table 2-42 scope-template Command Attributes (continued)

Attribute	Usage	Description
bootp	set get unset	Controls whether the server accepts BOOTP requests. If you want clients to always receive the same addresses, you need to reserve IP addresses for all your BOOTP clients. Optional, default disable.
deactivated	enable disable	A scope that does not extend leases to clients. It treats all of the addresses in its ranges as if they were individually deactivated. Optional, no default.
dhcp	**enable** disable	Controls whether the DHCP server accepts DHCP requests for this scope. Disable DHCP for a scope if you want it to use BOOTP exclusively, or to temporarily de-activate it. Optional, default enable.
dns-host-bytes	set get unset	Number of the bytes in a lease's 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. Optional, no default.
dynamic-bootp	set get unset	Number of the bytes in a lease's 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. Optional, no default.
embedded-policy	get	Embedded policy for the scope. Read-only. This attribute gets its value from the **scope-policy** command.
free-address-config	set get unset	SNMP trap configuration (see the "addr-trap" section on page 2-7). Optional, no default.
grace-period	set get unset	The length of time between the expiration of a lease and the time it is made available for re-assignment. Optional, no default.
ignore-declines	enable **disable** unset	Determines whether the scope should turn off recognition of server lease declines. Optional, default disable.
name	set get unset	The name of this scope template.
offer-timeout	set get unset	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. Optional, no default.
options-expr	set get unset	An expression to define the list of embedded policy options to be created for a scope object. Optional, no default.
ping-clients	enable disable	Controls whether the server should attempt to ping addresses before offering leases. Optional, no default.

Table 2-42 scope-template Command Attributes (continued)

Attribute	Usage	Description
ping-timeout	set get unset	The number of milliseconds that the DHCP server should wait for ping responses. If you make this value too large, you slow down the lease offering process. If you make this value too small, you reduce the effectiveness of pinging addresses before offering them. Three hundred milliseconds is a frequent compromise. Optional, default 300 ms.
policy	set get unset	Name of the policy associated with the scope. Required, default is the default policy. This means that the scope uses all the properties set in the default policy (including the lease time), unless specifically reset.
ranges-expr	set get unset	An expression to define the list of scope ranges to be created for a scope object. Optional, no default.
renew-only	set get unset	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). Optional, no default.
router-host	set get unset	Creates the IP address from the subnet to put it on the router interface because a router interface does not take a network ID. It takes an IP address and mask. Optional, no default.
selection-tag-list	set get unset	The list of selection tags to associate with a scope. Optional.
update-dns-for-bootp	enable disable unset	If the server replies to a BOOTP request and offers a lease from a scope that is configured to perform DNS updates, it checks this attribute before beginning the DNS update. This attribute prevents DNS updates for BOOTP clients while allowing updates for DHCP clients. You can also control this attribute globally using the **dhcp enable/disable update-dns-for-bootp** command, but the scope setting overrides it. Optional, no default.
vpn-id	get	The ID of the dhcp vpn that contains this scope. The vpn-id of a scope is initialized when the scope is created and cannot be edited once it is set.

Related Commands **dhcp**, **scope**

scope-template-policy

Use the **scope-template-policy** command to edit a DHCP policy that is embedded in a scope-template. An embedded policy is a collection of DHCP option values and settings associated with 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.

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*

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*

scope-template-policy *name* **getVendorOption** *opt-name* | *id* [*opt-set-name*]

Attributes

Table 2-43 describes the **scope-template-policy** command attributes.

Table 2-43 scope-template-policy Command Attributes

Attribute	Usage	Description
affinity-period	set get unset	For DHCPv6, specifies for how long a lease that has become available is retained for a client before it is deleted. This allows a client to obtain an expired lease if the client returns during this period; or it can prevent a client from reusing an address if it returns within this period (if either inhibit-all-renews or inhibit-renews-at-reboot is enabled). Optional, no default.
allow-client-a-record-update	enable **disable**	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. Optional, default is FALSE.

Table 2-43 scope-template-policy Command Attributes (continued)

Attribute	Usage	Description
allow-client-hints		If allow-client-hints is true, addresses and prefixes requested by the client, in SOLICIT and REQUEST messages, are used if possible. If allow-client-hints is false, addresses and prefixes requested by the client are ignored. Optional, the default is false.
allow-dual-zone-dns-update	enable **disable**	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. Optional, the default is FALSE.
allow-lease-time-override	enable **disable**	Indicates that clients may request a specific lease-time. The server will not honor those requested lease-times if this attribute is set to false. The server will not honor a client's lease-time if that time is longer than the server's normal lease-time. Optional, default is disabled.
allow-non-temporary-addresses	**enable** disable	Determines whether DHCPv6 clients can request non-temporary addresses. Optional, default is TRUE.
allow-rapid-commit	enable **disable**	Determines whether DHCPv6 clients can use a 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. This attribute needs special handling in processing the policies. The server checks all prefix policies (both embedded and named) for the link to which the client has access: • 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. Optional, default is FALSE.
allow-temporary-addresses	**enable** disable	Determines wether DHCPv6 clients can request temporary addresses.
default-prefix-length	set get unset	For delegation, specifies the default length of the delegated prefix if it is not explicitly requested by the requesting router (client). The default length must always be less than or equal to the prefix length of the prefix range. Optional, default is 64.
forward-dnsupdate	set get unset	Specifies the forward zone DNS update. Optional, no default.
forward-zone-name	set get unset	Names an optional forward zone to update. Optional, no default.

Table 2-43 scope-template-policy Command Attributes (continued)

Attribute	Usage	Description
giaddr-as-server-id	enable **disable**	Enables the DHCP server to set the server-id option on a DHCPOFFER and a DHCPACK to the giaddr of the incoming packet, instead of the IP address of the server (as it will by default). This cases 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 may not actually be an address to which the DHCP client can unicast a packet. In these cases, the DHCP client cannot renew a lease, and must always performing a rebind operation (where the DHCP client broadcasts a request instead of unicasting it to what it believes is the DHCP server). This feature is disabled by default.
grace-period	set get unset	Defines the length of time between the expiration of a lease and the time it is made available for re-assignment. Optional, default is 5m.
inhibit-all-renews	enable **disable**	Permits clients to renew their leases, but the server will force them to obtain new addresses each time they reboot. Optional, default is FALSE.
inhibit-renews-at-reboot	enable **disable**	Permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot. Optional, default is FALSE.
limitation-count	set get unset	Specifies the maximum number of clients with the same limitation-id that are allowed to have currently active leases. Optional, no default.
longest-prefix-length	set get unset	For delegation, the longest allowable length for prefixes. If the requesting router (client) requests a prefix length that is longer than this, the value set in this attribute is used instead. Optional, The default is the default-prefix-length.
offer-timeout	set get unset	Tells the server to wait the specified amount of time if it has offered a lease to a client but the offer is not accepted. At the end of the specified time interval, the server makes the lease available again. Optional, default is 2m.
packet-file-name	set get unset	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. Optional, no default.
packet-server-name	set get unset	Identifies the host-name of a server to used 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. Optional, no default.
packet-siaddr	set get unset	Identifies the IP address of the next server in a client's 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. Optional, no default.
permanent-leases	enable **disable**	Indicates that leases for this scope should be permanently granted to requesting clients. Optional, default is disabled.

Table 2-43 scope-template-policy Command Attributes (continued)

Attribute	Usage	Description
preferred-lifetime	set get unset	Specifies the default and maximum preferred lifetime for leases to DHCPv6 clients. Optional, default value is 1w (week).
reverse-dnsupdate	set get unset	Specifies the reverse zone DNS update. Optional, no default.
server-lease-time	set get unset	Tells the server for how long a lease is valid. For more frequent communication with the client, it may be useful to have the server consider leases leased for a longer period than the client does. 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. Optional, no default.
shortest-prefix-length	set get unset	For delegation, The shortest prefix length allowed for delegated prefixes. If the requesting router (client) requests a prefix length that is shorter than this, the value set in this attribute is used instead. Optional, the default is the default-prefix-length.
split-lease-times	enable **disable**	If enabled, the DHCP server uses the value of the *server-lease-time* attribute internally. Clients are still offered lease times that reflect the configured lease-time option from the appropriate policy, but the server bases its decisions about expiration on the *server-lease-time* value. Optional, default is disabled.
unavailable-time out	set get unset	Permits the server to make a lease unavailable for the specified period of time and then to return the lease available state. Optional. 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	enable **disable**	When checking the server's database for IP addresses that reserved, the server by default uses the MAC address of the DHCP client as the key for the database lookup. If *use-client-id-for-reservations* is enabled, then the check for reserved leases is performed by using the client-id of the DHCP client. The client-id is usually supplied by the DHCP client. In cases where it was not supplied by the DHCP client, then it is synthesized by the server, and that value will be used. Optional, the default is disabled.
v4-bootp-reply-options	set get unset	Lists the options that are returned to all BOOTP clients, whether or not a client specifically asks for the option data. Optional, no default.
v4-reply-options	set get unset	Lists the options that are returned to all DHCPv4 clients, whether or not a client specifically asks for the option data. Optional, no default.
v6-reply-options	set get unset	A list of options that should be returned in any replies to DHCPv6 clients.
valid-lifetime	set get unset	Specifies the default and maximum valid lifetime for leases to DHCPv6 clients. Optional, default value is 2w (weeks).

Usage Guidelines

You can set individual option values with the **setOption** command, unset option values with the **unsetOption** command, and view option values with the **getOption** and **listOptions** commands. When you set an option value the DHCP server will replace any existing value or create a new one as needed for the given option name. See the help file for the policy command for a complete description.

Related Commands

policy, client-policy, client-class-policy, dhcp-address-block-policy, dhcp-link-policy, dhcp-prefix-policy, scope-policy

server

The **server** command affects the behavior of the DNS, DHCP, or TFTP server. Any time you change the server configuration, use the **reload** command.

Timesaver The **server** keyword is optional. You can enter all these commands starting with just the server type.

[**server**] {**dns** | **dhcp** | **tftp**} **enable** [**start-on-reboot**]

[**server**] {**dns** | **dhcp** | **tftp**} **disable** [**start-on-reboot**]

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **start**

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **stop**

[**server**] {**dns** | **dhcp** | **tftp**} **get version**

[**server**] {**dns** | **dhcp** | **tftp**} **getHealth**

[**server**] {**dns** | **dhcp** | **tftp**} **getStats**

[**server**] {**dns** | **dhcp** | **tftp**} **reload**

[**server**] **dhcp getRelatedServers** [**column-separator=***string*]

[**server**] **dhcp setPartnerDown** *partner-server* [*date*]

[**server**] **dhcp updateSms** [**all**]

[**server**] {**dns** | **dhcp** | **tftp**} **serverLogs nlogs=***value* **logsize=***value*

[**server**] {**dns** | **dhcp** | **tftp**} **serverLogs show**

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **setDebug** *category*=*level* [*output*]

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **unsetDebug**

Syntax Description

The syntax descriptions use the convention {**dns** | **dhcp** | **tftp**} to indicate that you can use the command with the DNS, DHCP, or TFTP servers. There are no attributes other than those specified in the syntax. You can omit the **server** keyword in each case.

[**server**] {**dns** | **dhcp** | **tftp**} **enable** [**start-on-reboot**]

Enables a server. With the *start-on-reboot* attribute, the Server Agent starts the server when you reboot. You might want to disable this for clusters that provide a single protocol service. By default, the DNS and DHCP servers are enabled, while the TFTP server is disabled, to start on reboot.

[**server**] {**dns** | **dhcp** | **tftp**} **disable** [**start-on-reboot**]

Disables a server or the optional *start-on reboot* attribute. See the **enable** syntax.

```
nrcmd> dns disable start-on-reboot
```

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **start**

Starts a server DNS, DHCP, SNMP, or TFTP server.

[**server**] {**dns** | **dhcp** | **snmp** | **tftp**} **stop**

Stops a server (DNS, DHCP, SNMP, or TFTP). This does not terminate the server process, but stops it from handling further requests.

[**server**] {**dns** | **dhcp** | **tftp**} **get version**

Gets the version number of the server software. Useful when describing version information to the Cisco Technical Assistance Center (TAC).

[**server**] {**dns** | **dhcp** | **tftp**} **getHealth**

Gets the current health of a server. 0 indicates that the server is not running. For the DHCP server, 1 through 10 indicate how well the server is running—10 indicates the highest health. If there is an incremental drop in the server health value, look at the log files for the server as the best indication of health. The DNS and TFTP servers return values of either 0 (not running) or 10 (running).

[**server**] {**dns** | **dhcp** | **tftp**} **getStats**

Retrieves statistics from a running server. You supply one or more specific categories of statistics counters, or the keyword **all** to retrieve all.

[**server**] {**dns** | **dhcp** | **tftp**} **reload**

Stops and immediately restarts the server. When the server restarts, it rereads all of its configuration information and its previously saved state information and then begins operating.

[**server**] **dhcp getRelatedServers** [**column-separator**=*string*]

Gets the status of the connection between the DHCP server and its DNS, LDAP, or failover servers. You can optionally specify that the report use *string* for column separators.

[**server**] **dhcp setPartnerDown** *partner-server* [*date*]

Notifies the DHCP server that its partner DHCP server is down and moves all appropriate scopes into the PARTNER-DOWN state. Optionally, you can specify the date and time when the partner was last known to operate. The default is the current date.

Caution Ensure that the partner server is really down before using the **setPartnerDown** keyword.

[**server**] **dhcp updateSms** [**all**]

Causes the DHCP server to perform System Management Server (SMS) network discovery. Optionally, including *all* sends out all leased addresses from the DHCP server to SMS. Omitting *all* sends only those addresses leased since the last time you used this command.

[**server**] {**dns** | **dhcp** | **tftp**} **serverLogs nlogs**=*value* **logsize**=*value*

Sets or changes **nlogs**, the number of server logs, and **logsize**, the size of the server logs in bytes for a server. Valid values for **nlogs** are 2 through 100. The value of **logsize** is in bytes, and the optional K and M suffixes scale the specified value by 1000 or 1,000,000, respectively. Valid values for **logsize** are 10000 through 500000000 (or 10K through 500M) bytes. This example sets the DNS server to generate up to seven log files of five million bytes each. Restart the Network Registrar Server Agent for the changes to take effect:

```
nrcmd> dns serverLogs nlogs=7 logsize=5M
nrcmd> exit
(UNIX)> /etc/init.d/nwreglocal start
(Windows)> net start "Network Registrar Local Server Agent"
```

[server] {dns | dhcp | tftp} serverLogs show

Displays the number and size of log files.

[server] {dns | dhcp | snmp | tftp} setDebug *category=level* [*output*]

Sets the debugging level and debug message output location. See Table 2-44 for the most commonly used server debug category codes and levels. The debug details increase as you increase the level number. The valid output values are MLOG (the default), FILE *file*, and WINDOW. If you reload the DNS server after enabling the debug settings through the Web UI, Network Registrar disables debug—you must re-enable the debug setting.

Caution Setting the debug level can have a serious impact on your system performance. In addition to affecting performance, debug settings persist across reload events. Contact the Cisco Technical Assistance Center (TAC) first before using it.

[server] {dns | dhcp | snmp | tftp} unsetDebug

Unsets debugging for the server.

Table 2-44 Server Debug Category Codes

Server	Category	Level	Description
DNS	D	1–6	Server initialization, forwarding, server generated queries, incremental and full zone transfer requests and responses.
	U	1–2	DNS updates.
	N	1–5	NOTIFY packets.
	P	2–3	DNS packets.
DHCP	VX	1	Incoming and outgoing detailed packet tracing.
	KP	1–9	DNS update packet tracing and full details on all messages to and from LDAP, including all attribute values.
	Q	1–9	Client-class tracing.
	Y	1–4	Failover tracing.
SNMP	M		Main module, generic tracing.
	D		Database activity tracing.
	S		SCP client/server activity tracing.
	L		Listen module (select loop) tracing.
	S		SNMP module tracing.
	C		Cache module tracing.
TFTP	E	1–5	CSRC 1.0 extension object.
	F	1–5	File handling.
	C	1–5	Server configuration.
	S	1–5	TFTP session handling.
	D	1–5	Statistics.
	P	1–5	Packet handling.
	T	1–5	Timer handling.

Usage Guidelines

Starting and Stopping Servers

Use the **server** *type* **start** command (or simply *server-type* **start**, such as **dhcp start**) to start the specified server. Use the **server** *type* **stop** (or simply *server-type* **stop**, such as **dhcp stop**) command to stop the specified server. It is advisable to save the server first:

```
nrcmd> dns start
nrcmd> save
nrcmd> dhcp stop
```

Reloading Servers

Use the **server** *type* **reload** (or simply *server-type* **reload**) command to reload the specified server. Network Registrar stops the server you chose, loads the configuration data, and then restarts the server:

```
nrcmd> dns reload
```

Logging Server Events

The DNS, DHCP, and TFTP servers have log settings that can severely restrict what is logged, and thereby improve server performance. These log settings are available using the **dns set log-settings**, **dhcp set log-settings**, and **tftp set log-settings** commands in the CLI, respectively.

Caution To avoid filling up the Windows Event Viewer and preventing Network Registrar from running, in the Event Log Settings, check the **Overwrite Events as Needed** box.

You can find out how the server maximums are set using the [**server**] *type* **serverLogs show** command, looking at the number (*nlogs*) and size (*logsize*) parameters, and changing them if necessary:

```
nrcmd> dhcp serverLogs show
nrcmd> dhcp serverlogs nlogs=6 logsize=200000
```

After making changes, stop and restart the Server Agent. On:

- Windows:

```
net stop "Network Registrar Local Server Agent"
net start "Network Registrar Local Server Agent"
```

- Solaris:

```
/etc/init.d/nwreglocal stop
ps -leaf | grep nwr
kill -9 pid pid ... any processes left running in this ps ....
/etc/init.d/nwreglocal start
```

- Linux:

```
/etc/rc.d/init.d/nwreglocal stop
ps -leaf | grep nwr
kill -9 pid pid ... any processes left running in this ps ....
/etc/rc.d/init.d/nwreglocal start
```

Displaying the Server's Health

To display a server's health (how well it is running), use the [**server**] *type* **getHealth** command. The number 10 indicates the highest level of health, and 0 indicates that the server is not running.

Getting Server Statistics

Use the [**server**] *type* **getStats** command to get statistics for the server. The statistics for the DNS and DHCP servers are encoded in curly braces followed by sets of digits. See the *Network Registrar User's Guide* for descriptions of these statistics.

Related Commands **dns**, **dhcp**, **snmp**, **tftp**

session

The **session** command sets session control parameters on your CLI command session.

The session assert functionality allows a nrcmd batch script to assert that a given condition is true. If the condition is true, the command has no effect, but if it is not true, the **nrcmd** exits at that point. Some uses for the session assert functionality are to ensure that the **nrcmd** session has an exclusive lock on the Network Registrar database, or to verify whether or not server configuration data has changed since a previous point.

session set *attribute*

session unset *attribute*

session get *attribute*

session [**show**]

session cache {**refresh** | **clear**}

session listNetInterfaces

Syntax Description

session set *attribute*

Sets one or more of the attributes in Table 2-45 on page 2-174.

session unset *attribute*

Unsets one or more of the attributes, if possible, in Table 2-45 on page 2-174.

session get *attribute*

Displays one of the attributes in Table 2-45 on page 2-174.

session [**show**]

Shows the values of all attributes assigned to the CLI session.

session cache {**refresh** | **clear**}

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.

session listNetInterfaces

Retrieves the list of network interface objects from the CCM server.

Attributes

Table 2-45 describes the attributes for the **session** command.

Table 2-45 session Command Attributes

Attribute	Usage	Description
cluster	get	Shows the name of the current cluster. Read-only; cannot be unset.
current-vpn	set get unset	Sets the VPN for the session, if there is a VPN expected for a CLI command, but that command does not have an explicit entry for the VPN or you cannot explicitly enter it for the command. If omitted, the global VPN is used. The VPN specified can be its name or ID. The special VPN value **all** refers to all VPNs, including global, and the special value **global** refers to the global VPN with no name assigned. If the string matches an already defined VPN, it is considered a VPN name; it is otherwise considered a VPN ID and the CLI tries to convert it to one. See the "owner" section on page 2-129. Unsetting the current VPN is the equivalent of the **session set current-vpn=""** command.
default-format	set get	Sets the default output format for the CLI session, with the output content based on the visibility level set (see the *visibility* attribute). You cannot unset this attribute. The default output formats are: • **user**—Show objects in a user-readable format, one attribute per line (the default). • **script**—Show objects in script-suitable format, one object per line. Unlike user format, this output does not display attributes that are not assigned values. This example sets the output for script processing: `nrcmd> session set default-format=script`
groups	show	Displays the list of groups associated with the current user.
roles	show	Displays the list of administrator roles associated with the current user.
scope-edit-mode	set show	Edit mode currently in effect when editing DHCP scopes. The valid values are: • **default**—Sets the scope edit mode configured at the CCM server (see ccm, page 2-12), the default. • **staged**—Determines if edits are written to the database, but not immediately forwarded to the DHCP server. This setting overwrites the CCM server setting. • **synchronous**—Determines if edits are immediately forwarded for publishing to the DHCP server. This setting overwrites the CCM server setting.
user-name	get	Shows the name of the current user. Read-only; cannot be unset.
version	get	Shows the software version of the cluster. Read-only; cannot be unset.

Table 2-45 session Command Attributes (continued)

Attribute	Usage	Description
visibility	set get	Session visibility, or what verbosity of attributes you can set and display. The valid values are **1** (highest visibility), **3**, or **5** (the default). You cannot unset this attribute. **Caution** Do not change the default session visibility from 5 unless directed by the Cisco Technical Assistance Center (TAC).
zone-edit-mode	set show	Edit mode currently in effect when editing DNS zones. The valid values are: • **default**—Sets the zone edit mode configured at the CCM server (see zone, page 2-196), the default. • **staged**—Determines if edits are written to the database, but not immediately forwarded to the DNS server. This setting overwrites the CCM server setting. • **synchronous**—Determines if edits are immediately forwarded for publishing to the DNS server. This setting overwrites the CCM server setting.

Related Commands **dns**, **dhcp**

snmp

Use the **snmp** command to control and configure the Simple Network Management Protocol (SNMP) server. Attributes you can set include log settings, whether to use configured rather than discovered interfaces, listening on a nonstandard port number, and cache time-to-live.

For SNMP traps, see the "trap-recipient" section on page 2-188.

snmp start

snmp stop

snmp enable server-active

snmp disable server-active

snmp set *attribute*=*value* [*attribute*=*value*…]

snmp unset *attribute*

snmp get *attribute*

snmp setDebug

snmp [**show**]

Syntax Description

See Table 2-46 on page 2-177 for the **snmp** command attributes and their descriptions.

snmp start

Starts the SNMP server.

snmp stop

Stops the SNMP server.

snmp enable server-active

Enables the server to run when started.

snmp disable server-active

Disables the server from running when started.

snmp set *attribute*=*value* [*attribute*=*value*]

Sets one or more SNMP server attributes.

snmp unset *attribute*

Unsets one or more SNMP server attributes.

snmp get *attribute*

Gets an SNMP server attribute.

snmp setDebug

Sets debugging for the SNMP server.

snmp [**show**]

Shows the attributes for the SNMP server.

Attributes

Table 2-46 describes the **snmp** command attributes and their values. They are all read-only attributes.

Table 2-46 snmp Command Attributes

Attribute	Usage	Description
community	set get unset	Community string required in the request for the server to process the request. Optional, default public.
cache-ttl	set get unset	Amount of time that data should remain in cache as the SNMP server is responding to GETs. Optional, default 60s.
log-settings	set get unset	Level of server activity logging, which can be one of: • **1—default** • **2—no-success-messages** • **3—incoming-packet-detail** • **5—outgoing-packet-detail** • **6—scp-detail** • **7—snmp-detail** Optional, default is the default setting.
name	set get unset	Name of the SNMP server. Optional, default CNRSNMP.
server-active	**enable** disable unset	Sets whether to enable the SNMP server to run when started. Optional, default enable.
trap-source-addr	set get unset	Optional IP address to use as the sender address in outgoing SNMP trap packets. Optional, no default.

Related Commands

server, **trap-recipient**

snmp-interface

The **snmp-interface** command adds, removes, and lists SNMP interfaces. An SNMP interface is a logical representation of the hardware interface (such as a server's Ethernet or Token Ring network interface card). SNMP uses the configured address information to determine which interface to use to send and receive packets. SNMP automatically discovers its interfaces and the list of available addresses on those interfaces.

snmp-interface *name* **create address=***IP/mask*

snmp-interface *name* **delete**

snmp-interface *name* **enable address**

snmp-interface *name* **disable address**

snmp-interface *name* **show**

snmp-interface *name* **set address=***IP/mask*

snmp-interface *name* **unset address**

snmp-interface *name* **get address**

snmp-interface list

snmp-interface listnames

Syntax Description

See Table 2-46 on page 2-177 for the **snmp-interface** command attributes and their descriptions.

snmp-interface *name* **create address=***IP/mask*

Creates an SNMP interface and optionally assigns attribute values. The name and attribute values are required for this command.

snmp-interface *name* **delete**

Deletes an snmp interface.

snmp-interface *name* **enable address**

Enables an attribute on the specified snmp interface.

snmp-interface *name* **disable address**

Disables an attribute on the specified snmp interface.

snmp-interface *name* **set address=***IP/mask*

Sets an attribute to a value for an snmp interface.

snmp-interface *name* **unset address**

Unsets the value of an snmp attribute.

snmp-interface *name* **get address**

Gets the explicit value of an attribute for the specified snmp interface.

snmp-interface *name* [**show**]

Shows the values of all attributes assigned to the snmp interface.

snmp-interface list

Lists all SNMP interfaces and any attributes assigned to them.

Attributes

The snmp-interface command has these attributes:

- address—The IP address and subnet mask of the interface that the SNMP server should use. Optional, no default.
- name—The name of the interface. Optional, no default.

Related Commands

snmp

subnet

The **subnet** command creates and sets attributes for network subnets created in the Network Registrar Central Configuration Management (CCM) database. A subnet is a contiguous range of IP address space that can be parented by an address block.

Note A CCM subnet is not the same as a DHCP subnet used for delegation to DHCP servers in virtual private network (VPN) and subnet allocation deployments. You manage these DHCP subnets using the **dhcp-subnet** command.

subnet *address/mask* **create** [*attribute*=*value* ...]

subnet *address/mask* **delete**

subnet *address/mask* **set** *attribute*=*value* [*attribute*=*value* ...]

subnet *address/mask* **unset** *attribute*

subnet *address/mask* **get** *attribute*

subnet *address/mask* [**show**]

subnet list

subnet listnames

Syntax Description

See Table 2-47 on page 2-181 for the **subnet** command attributes and their descriptions.

subnet *address/mask* **create** [*attribute*=*value* ...]

Creates a CCM subnet with a network address (in the *address/mask* format), and optionally adds attributes. The policy is the only required attribute, which defaults to *default* if omitted.

subnet *address/mask* **delete**

Deletes a CCM subnet.

subnet *address/mask* **set** *attribute*=*value* [*attribute*=*value* ...]

Sets one or more attributes for the CCM subnet. For example:

```
nrcmd> subnet 10.1.0.0/16 set vpn-id=vpn 1
```

subnet *address/mask* **unset** *attribute*

Unsets an optional CCM subnet attribute.

subnet *address/mask* **get** *attribute*

Gets the explicit value for a CCM subnet attribute.

subnet *address/mask* [**show**]

Shows the values of all attributes of the CCM subnet.

subnet list

Lists all CCM subnets and their attributes.

subnet listnames

Lists only the names of all CCM subnets.

Attributes

Table 2-47 describes the **subnet** command attributes and their values and defaults, if any.

Table 2-47 subnet Command Attributes

Attribute	Usage	Description
address	create set get	IP address of the CCM subnet, specified at creation, defining its address range. Use the **set** command to redefine the address. Required, no default.
description	set get unset	Description of the use of the CCM subnet. Optional, no default.
dns-host-bytes	set get unset	Parallels the scope *dns-host-bytes* attribute, and used when creating reverse zones from a subnet to determine the correct in-addr.arpa name to create. Based on its value and the subnet size, either a new reverse zone is created or delegation records are put in the parent reverse zone. If unset, the server synthesizes an appropriate value based on the subnet size.
failoverpair	set get unset	DHCP server failover pair or cluster assigned to the subnet for DHCP allocation. Optional, no default.
forward-zone-name	set get unset	Name of the forward DNS zone associated with the subnet. Optional, no default.
interface	set get unset	Router interface assigned to this subnet. Optional, no default.
owner	set get unset	Name of the owner possibly used to limit access to the subnet. Optional, no default.
parent	set get unset	The parent CCM address block of the subnet, if any. Optional, no default.
primary-subnet	set get unset	Network number of the primary subnet, when multiple logical IP subnets are present on the same physical network. Optional, no default.
region	set get unset	Name of the region possibly used to limit access to the subnet. Optional, no default.
reverse-zone-name	set get unset	Name of the reverse DNS zone associated with the subnet. Optional, no default.
state	get	Current state of the subnet. Read-only.
type	set get unset	Name of the defined type of the subnet, if to be associated with a scope template, scope-selection tag, or client-class. Optional, no default.

Table 2-47 subnet Command Attributes (continued)

Attribute	Usage	Description
vpn-id	set get unset	ID of the virtual private network (VPN) used to support multiple address spaces, such as in a managed VPN environment. Optional, no default.

Related Commands **policy,client-policy,client-class-policy, dhcp-address-block-policy, dhcp-link-policy, scope-policy, scope-template-policy**, **address-block**, **owner**

tftp

The **tftp** command enables or disables TFTP server attributes. Because there is only one TFTP server per cluster within Network Registrar, you do not need to refer to the server by name.

tftp enable *attribute*

tftp disable *attribute*

tftp set *attribute*=*value* [*attribute*=*value*…]

tftp unset *attribute*

tftp get *attribute*

tftp [**show**]

tftp setTraceLevel *value*

tftp getTraceLevel

tftp reload

Note See also the "server" section on page 2-168 for other server control commands.

Syntax Description

See Table 2-48 on page 2-184 for the **tftp** command attributes and their descriptions.

tftp enable *attribute*

Enables a TFTP server attribute.

```
nrcmd> tftp enable file-caching
```

tftp disable *attribute*

Disables a TFTP server attribute.

tftp set *attribute*=*value* [*attribute*=*value*…]

Sets one or more attributes of the TFTP server (see Table 2-48 for the attributes with a usage of set):

```
nrcmd> tftp set file-cache-directory="CacheDir"
nrcmd> tftp reload
```

Note If you use this command, you must set the cache directory and reload the server. If *file-cache* is enabled, but *file-cache-directory* is not set, no files are cached. Even if *file-cache* is disabled, but *file-cache-directory* is set, the files in the directory are still accessible to clients.

tftp unset *attribute*

Unsets the value of an attribute of the TFTP server. You cannot unset required attributes.

tftp get *attribute*

Gets the explicit value of an attribute for the TFTP server.

tftp [**show**]

Shows the TFTP server attributes.

tftp setTraceLevel *value*

Level of tracing that the TFTP server uses. Trace output is written to the file_tftp_1_log file in the server's logs directory. Trace statements go to the file_tftp_1_log on Windows NT and to the file_tftp_1_trace file on Solaris. Each integer value from 0 through 4 enables another cumulative trace level:

- 0—Disables all server tracing (default).
- 1—Displays all server log messages in the trace file.
- 2—Also displays the client IP address and port for all TFTP packets.
- 3—Also displays header information for all TFTP packets.
- 4—Also displays the first 32 bytes of TFTP packet data.

Note Only enable packet tracing if the Cisco TAC instructs you to. Tracing has significant impact on the performance level of the server. Also, do not enable packet tracing for long periods of time.

tftp getTraceLevel

Reports the trace level. Use only when investigating server problems.

tftp reload

Reloads the TFTP server and updates the files in cache.

Attributes

Table 2-48 describes the **tftp** command attributes and their values and defaults, if any.

Table 2-48 tftp Command Attributes

Attribute	Usage	Description
active-directory-domain	set get	Name of an active directory domain that the TFTP server uses to provide dynamic configuration file support. Required, no default.
csrc-configuration-file	set get	Path to a configuration file the TFTP server uses when loading the Cisco Subscriber Registration Center (CSRC) version 1.0 library. The TFTP server can then generate dynamic DOCSIS modem configuration files. The location of the CSRC configuration file is typically /CSRC_INSTALL_DIR/conf/csrc.cfg. Required, no default.
default-device	set get	Name of the default disk device the TFTP server uses when none is specified in the pathname in the TFTP request. This attribute is designed for use on Windows to specify a default drive letter. Required, no default.
docsis-access	enable **disable**	How the TFTP server should respond to dynamic DOCSIS file requests from TFTP clients. Relevant for users of CSRC 1.0 only. If this attribute is disabled, the TFTP server refuses dynamic DOCSIS file requests and sends an access violation error to the client. Required, default disable.
docsis-file-logging	enable **disable**	Whether the TFTP server should log generated DOCSIS files to disk. Relevant for users of CSRC 1.0 only. If this attribute is enabled, the TFTP server logs each generated DOCSIS configuration file to a tftp subdirectory within the server logs directory. Relevant for users of CSRC 1.0 only. Required, default disable.

Table 2-48 tftp Command Attributes (continued)

Attribute	Usage	Description
docsis-log-file-count	set get	Maximum number of DOCSIS configuration log files that the TFTP server maintains in the TFTP subdirectory within the server logs directory. Relevant for users of CSRC 1.0 only. Once this limit is reached, the TFTP server removes one DOCSIS log file for each new log file it creates. Required, default 100 log files.
docsis-pathname-prefix	set get	Pathname prefix that the TFTP server recognizes as the trigger to create a DOCSIS configuration file. Relevant for users of CSRC 1.0 only. This prefix must match the one that the DHCP server uses to generate the DOCSIS filename sent to the TFTP client. Required, default /docsis.
file-cache	enable **disable**	Determines whether the TFTP server should perform file caching on files located in the directory that the *file-cache-directory* attribute specifies. File caching allows the server to run faster by loading the files into memory, up to the maximum set by the *file-cache-max-memory-size* attribute. Upon reload, Network Registrar logs the name of each cached file, and skips any files it cannot load. It reads in all files as binary data and translates them as the TFTP client requests. Writing directly to cache is not allowed. Optional, default disable.
file-cache-directory	set get	Path to an existing directory where the TFTP server finds the files to put into cache, if enabled by the *file-cache* attribute. The server loads all these files on startup and on reloading, up to the maximum set by the *file-cache-max-memory-size* attribute. Use pathnames relative to the value of the *home-directory* attribute only. Network Registrar does not cache any files in this directory's subdirectories. Optional, no default, but the value is appended to the home directory path.
file-cache-max-memory-size	set get unset	Maximum memory size, in bytes, of the file cache. Network Registrar loads all files into cache that cumulatively fit this memory size. If set to 0, Network Registrar does not cache any data, even if you enable file caching. Optional, default 32000 bytes.
home-directory	set get	Path to a home directory that the TFTP server uses to resolve TFTP requests. With the *use-home-directory-as-root* attribute disabled, Network Registrar uses the value of the *home-directory* attribute plus the paths specified in the search-list to resolve requests. Required, default is the /data/tftp subdirectory of the installation directory.
initial-packet-timeout	set get	Initial time that the TFTP server waits after sending a response to a client before declaring that response timed-out and sending a retransmission to the client. Required, default 5s.
ldap-host-name	set get	Host name or IP address of an LDAP server that the TFTP server uses to provide dynamic configuration file support. Relevant for users of CSRC 1.0 only. Required, default localhost.
ldap-initial-timeout	set get	Initial time the TFTP server waits after sending a request to an LDAP server before declaring that request timed-out and sending a retransmission to the server. Relevant for users of CSRC 1.0 only. Required, default 10s.
ldap-maximum-timeout	set get	Maximum time that the TFTP server waits after transmitting the initial LDAP request before giving up retrying on that request. Relevant for users of CSRC 1.0 only. Required, default 60s.

Table 2-48 tftp Command Attributes (continued)

Attribute	Usage	Description
ldap-password	set get	Password that the TFTP server uses when connecting to an LDAP server. Relevant for users of CSRC 1.0 only. Required, no default.
ldap-port-number	set get	Port number that the TFTP server uses when communicating with an LDAP server. Relevant for users of CSRC 1.0 only. Required, default 389.
ldap-root-dn	set get	Root distinguished name that the TFTP server uses to locate the root of the directory tree for dynamic configuration file support. Relevant for users of CSRC 1.0 only. Required, no default.
ldap-use-ssl	enable **disable**	Controls whether the TFTP server uses SSL when communicating with an LDAP server. Relevant for users of CSRC 1.0 only. If this attribute is disabled, the TFTP server does not use SSL when communicating with LDAP. Required, default disable.
ldap-user-name	set get	Username of the TFTP server when connecting to an LDAP server. Relevant for users of CSRC 1.0 only. Required, no default.
log-level	set get unset	Level of verbosity that the TFTP server employs when writing log messages to the TFTP server log file. Required, default level 3. Each integer value from zero through four enables these cumulative log levels. It is best to maintain the log level at the default of 3 (information): • **0**—None: no log messages written. • **1**—Error: present condition inhibits the TFTP server operation, such as there is no LDAP server. • **2**—Warning: present condition can cause operational problems, such as connection timeouts. Also includes errors. • **3**—Information: provides normal server informational messages (default). Also includes warnings and errors. • **4**—Activity: normal server operation, such as client requests and replies. Also includes information, warnings, and errors.
log-settings	set get unset	The TFTP server allows control over additional details about the events listed in the log settings.The additional detail can be very helpful when analyzing a problem, but can cause the log files to fill up quickly (and therefore to turn over frequently, possibly losing important information) if left enabled for a long period of time. Optional, default verbose.
max-inbound-file-size	set get	Maximum file size that the TFTP server enforces for a file written to the TFTP server. The default unit is kilobytes, but you can use k, m, or g to indicate kilobytes, megabytes or gigabytes. Required, default 1024 KB.
min-socket-buffer-size	set get	Minimum socket buffer size that the TFTP server uses for the well known port on which it is listening for TFTP requests. Required, default 65536.
packet-trace-level	set get	Specifies the level of verbosity that the TFTP server employs 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. Required, default 0 (disabled).
port-number	set get	UDP port number that the TFTP server uses to listen for TFTP requests. Required, default port 69.

Table 2-48 tftp Command Attributes (continued)

Attribute	Usage	Description
read-access	**enable** disable	How the TFTP server should respond to file read requests from TFTP clients. If this attribute is disabled, the TFTP server refuses file read requests and sends an access violation error to the client. Required, default enable.
search-list	set get	Comma-separated list of paths that the TFTP server uses to resolve TFTP requests. If you enable use-home-directory-as-root, the server ignores the paths in the search list and uses the home directory to resolve all TFTP requests. Required, no default.
session-timeout	set get	Maximum length of time that the TFTP server waits 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. Required, default 20s.
use-home-directory-as-root	enable **disable**	Whether the TFTP server treats pathnames contained in TFTP requests as if the paths were rooted at the specified home directory. If this attribute is enabled, the TFTP server attempts to resolve both absolute and relative pathnames to paths located beneath the specified home directory. Required, default disable.
write-access	enable **disable**	How the TFTP server should respond to file write requests from TFTP clients. If this attribute is disabled, the TFTP server refuses file write requests and sends an access violation error to the client. Limitations—If enabled, the client can only write to a file that already exists on the server. The *max-inbound-file-size* attribute dictates the incoming file size. Required, default disable.
write-allow-file-create	enable **disable**	Whether to allow file creation on a PUT. With *write-access* enabled, if this attribute is disabled, the file must exist on the server; if enabled, the file is created on the server. The *max-inbound-file-size* value sets the file size limit. Required, default disable.

Related Commands server

trap-recipient

The **trap-recipient** command enables you to add or delete destinations to which the Network Registrar SNMP server sends event messages to a specified a destination (probably a management station) for traps. The server attempts to send trap messages to all its configured recipients, which are usually network management stations.

trap-recipient *name* **create** [*attribute*=*value* ....]

trap-recipient *name* **delete**

trap-recipient *name* **set** *attribute*=*value*

trap-recipient *name* **unset** *attribute*=*value*

trap-recipient *name* **get** *attribute*

trap-recipient *name* **show**

trap-recipient list

Syntax Description

trap-recipient *name* **create** *attribute*=*value* ....

Creates a trap recipient and optionally assigns attribute values. The name and attribute values are required for this command.

trap-recipient *name* **delete**

Deletes the specified trap-recipient.

trap-recipient *name* **set** *attribute*=*value*

Sets the attributes and values on a specified trap recipient.

trap-recipient *name* **unset** *attribute*=*value*

Unsets an attribute and its value on a specified trap recipient.

trap-recipient *name* **get** *attribute*

Gets the explicit value assigned to an attribute.

trap-recipient *name* **show**

Shows all attributes and values assigned to a trap recipient.

trap-recipient list

Lists all configured trap recipients.

Attributes

Table 2-49 describes the **trap-recipient** command attributes.

Table 2-49 trap-recipient Command Attributes

Attribute	Usage	Description
agent-addr	set get unset	Defines the IP address used as the source agent address in traps that are sent to this recipient. Optional, no default.

Table 2-49 trap-recipient Command Attributes (continued)

community	set get unset	Defines the SNMP community string assigned to the trap recipient. Optional, default.
ip-addr	set get unset	Defines the IP address of the device that is the trap recipient. Optional, no default.
port-number	set get unset	Defines the IP port of the device that is the trap recipient. Optional, no default.

update-policy

The **update-policy** command lets you configure DNS update policies. The most significant attribute of an update policy is an ordered list of rules. The rules are used to restrict or permit updates and are used in the **zone** update-policy-list attribute.

An access control list (ACL) can be an IP address; network address; named ACL (see the "acl" section on page 2-2); or the reserved words **any**, **none**, **localhost**, or **localnets**. The optional index digit value is used to add the rule in a specific order, with lower digits taking priority.

update-policy *name* **create**

update-policy *name* **rules add "**{**grant** | **deny**} *acl* {**name** | **subdomain** | **wildcard**} *value rr-type***"** [*index*]

update-policy *name* **rules remove** *index*

update-policy *name* **delete**

Note If you have configured an update-acl on a zone, any update-policy configuration is ignored.

Syntax Description

update-policy *name* **create**

Creates a DNS update policy.

update-policy *name* **rules add "grant** *acl* **name** *value rr-type***"** [*index*]

Adds a DNS update rule to grant the client's matching ACL permission to update certain resource records based on their name and types.

update-policy *name* **rules add "deny** *acl* **name** *value rr-type***"** [*index*]

Adds a DNS update rule to deny the client's matching ACL permission to update certain resource records based on their name and types.

update-policy *name* **rules add "grant** *acl* **subdomain** *value rr-type***"** [*index*]

Adds a DNS update rule to grant the client's matching ACL permission to update certain resource records based either on their name and types, or on any other names in that subdomain with the same types.

update-policy *name* **rules add "deny** *acl* **subdomain** *value rr-type***"** [*index*]

Adds a DNS update rule to grant the client's matching ACL permission to update certain resource records based either on their name and types, or on any other names in that subdomain with the same types.

update-policy *name* **rules add "grant** *acl* **wildcard** *value rr-type***"** [*index*]

Adds a DNS update rule to grant the client's matching ACL permission to update certain resource records based on wildcards, which can be:

- *****—Matches zero or more characters. For example, the pattern **example*** matches all strings starting with *example*, including **example–**.
- **?**—Matches a single character only. For example, the pattern **example*.com** matches **example1.com** and **example2.com**, but not **example.com**.

- **[ ]**—Matches any characters in the brackets. The characters can be a range, such as **[0–9]** and **[a–z]**. If a pattern should include a hyphen, make it the first character, such as **example[–a–z]**.

update-policy *name* **rules add "deny** *acl* **wildcard** *value rr-type***"** [*index*]

Adds a DNS update rule to deny the client's matching ACL permission to update certain resource records based on wildcards. The wildcards are the same as listed under the **grant** syntax.

update-policy *name* **rules remove** *index*

Removes a DNS update rule based on its index.

update-policy *name* **delete**

Deletes a DNS update policy.

Attributes

Table 2-50 describes the **update-policy** command attributes.

Table 2-50 update-policy Command Attributes

Attribute	Description
name	The name of the update policy. Required, no default.
rules	The list of rules that make up the update policy. Optional, no default. Use the following syntax with each rule: `action acl-list keyword value rr-list` • action—**grant** or **deny** the ability to update if the update matches one of the rule conditions. • acl-list—Provide a list consisting of one or more IP addresses, network addresses, keys, and named acl references. Prefix key names with the reserved word **key**. • keyword—Identify how resource records (RR) should be interpreted to define the update policy. The choices are name, subdomain, and wildcard from the drop-down list: – name— The RR should have the specified name to match the rule. – subdomain—The RR should be, or be in, the specified subdomain to match the rule. – wildcard—The RR should match the wildcard value (but not its domain) to match the rule. The wildcards are *, matches zero or more characters; ?, matches a single character); [], matches any of the characters listed in the brackets (can also be indicated as a range, such as [a-z]). For each keyword you assign a value. • rr-list—A comma-separated list of RR types to match to the rules. You can negate each value by preceding it with an exclamation symbol (!). For example, you can explicitly deny updates of all RRs other than TXT and PTR records by selecting the deny Action, and entering a wildcard Keyword and !TXT,!PTR in the RR Types field. Required.

Usage Guidelines

Use rules to restrict or permit updates to DNS. When you add a new rule, enclose the complete string in quotes (“). Use the backslash (\) to escape the special bracket ([ ]) characters used in the rule. For example:

```
update-policy example create
update-policy example rules add "grant any wildcard ser* SRV"
update-policy example rules add "deny 10.10.10.1 wildcard \[a-z\]* A"
```

If an update does not match one of the rules, the final implicit rule is to deny any wildcard

Related Commands

acl, **zone**

vpn

The **vpn** command creates, deletes, sets, and lists attributes for virtual private networks (VPNs). A VPN distinguishes a set of DHCP server objects that are independent of otherwise identical objects in other VPNs. The DHCP server groups DHCP address blocks and their associated subnets, and scopes and their associated leases by VPN. A VPN has a descriptive name.

vpn *name* **create** *id* [*attribute*=*value*]

vpn *name* **delete**

vpn *name* **set** *attribute*=*value*

vpn *name* **unset** *attribute*

vpn *name* **get** *attribute*

vpn *name* [**show**]

vpn list

vpn listnames

Note The term VPN is synonymous with namespace, a term that Network Registrar software used for the management of clusters in versions of the product prior to 6.2.

There are two ways to use VPNs:

- Through DHCP address blocks—By creating subnets
- Through scopes—By creating leases

Syntax Description

vpn *name* **create** *id* [*attribute*=*value*]

Creates a VPN using a unique name and unique VPN identifier. The VPN requires the *name* and an *id*. You cannot use the reserved words **all** or **global** for the VPN name.

The VPN can take two attributes, the VPN Routing and Forwarding instance (VRF) name and the VPN identifier (See Table 2-51). Network Registrar associates an incoming packet with the VPN if either the VRF name or VPN ID appears in the vpn-id option or vpn-id suboption (each can carry only one at a time in the packet).

You can change the VPN name using the **set** command, unless the VPN is the current VPN defined by the **session set current-vpn** command or the new name is not unique. You cannot change the *vpn-id* value (see the "session" section on page 2-173).

vpn *name* **delete**

Deletes a VPN.

vpn *name* **set** *attribute*=*value*

Changes the VPN name or sets one of the other attributes (see Table 2-51). You can change the VPN name only to another unique name. You cannot change the *vpn-id* value.

vpn *name* **unset** *attribute*

Unsets the value of an attribute.

vpn *name* **get** *attribute*

Gets the explicit value of an attribute.

vpn *name* [**show**]

Shows the values of all attributes assigned to the VPN.

vpn list

Lists the vpns and their properties.

vpn listnames

Lists just the names of all the VPNs.

Attributes

Table 2-51 describes the **vpn** command attributes and *attribute*=*value* pairs.

Table 2-51 vpn Command Attributes

Attribute	Usage	Description
addr-blocks-default-selection-tags	set get unset	Specifies the default selection tag (or list of tags) that are associated with incoming subnet allocation requests in this VPN that does not contain any subnet name data. Optional, no default.
addr-blocks-use-client-affinity	enable disable unset	The DHCP server attempts to allocate subnets to clients using DHCP address blocks that the clients have already used. Use this parameter to disable that behavior, in which case the server supplies subnets from any DHCP address block that is suitable (based on other selection data in the clients' messages). Optional, no default.
addr-blocks-use-lan-segments	enable disable unset	Controls whether DHCP subnet allocation uses the *lan-segment* attribute when configured on DHCP address blocks. Optional, no default.
addr-blocks-use-selection-tags	enable disable unset	Controls whether the server compares incoming DHCP subnet allocation requests' subnet name data with each DHCP address block's selection tags. A DHCP address block is only considered if the two match. Optional, no default.
description	set get unset	Description for the VPN. Optional, no default.
id	create set get	Unique identifier for the VPN. Must be a positive number. Required, no default.
name	create set get unset	Unique name string for the VPN; for example, Red or Blue. Required, no default.

Table 2-51 vpn Command Attributes (continued)

Attribute	Usage	Description
vpn-id	set get unset	Unique identifier associated with the VPN. The VPN ID is in the format *oui*:*index*, for example a1:3f6c. It consists of a three-octet VPN authority organizationally unique identifier (OUI), assigned by the IEEE organization (RFC 2685), that corresponds to the VPN owner or ISP, followed by a colon, and a four-octet index number corresponding to the VPN serviced by the authority. DHCP and the Remote Authentication Dial-In User Service (RADIUS) use the VPN ID to identify a VPN. RADIUS can use it to assign dial-in users to the proper VPN, based on each user's authentication information. Optional, no default.
vrf-name	set get unset	Unique virtual routing forwarding (VRF) name, as derived from the relay agent router configuration. Optional, no default.

Related Commands **acl**, **dhcp**, **dhcp-address-block**, **policy,client-policy,client-class-policy, dhcp-address-block-policy, dhcp-link-policy, scope-policy, scope-template-policy**

zone

The **zone** command creates and edits DNS zones, as well as forces zone transfers.

zone *name* **create primary file=***hostfile* [*template=template-name*]

zone *name* **create primary** *nameserver person* [*template=template-name*] [*attribute=value …*]

zone *name* **create secondary** *address* [*attribute=value …*]

zone *name* **applyTemplate** *template-name*

zone *name* **delete**

zone *name* **enable** *attribute*

zone *name* **disable** *attribute*

zone *name* **set** *attribute=value* [*attribute=value ...*]

zone *name* **unset** *attribute*

zone *name* **get** *attribute*

zone *name* [**show**]

zone list

zone listnames

zone *name* **forceXfer secondary**

zone *name* **addHost** *hostname IPaddress* [*alias …*]

zone *name* **removeHost** *hostname*

zone *name* **listHosts**

zone *name* **addRR** [**staged | -sync**] *name* [*ttl*] [*class*] *type data*

zone *name* **removeRR** *name type data*

zone *name* **addDNSRR** *name* [*ttl*] *type data*

zone *name* **removeDNSRR** *name* [*ttl*] *type data*

zone *name* **protect-name | unprotect-name** *name*

zone *name* **listRR {all | ccm |dns}**

zone *name* [**protect-name | unprotect-name**] *name*

zone *name* **syncToDns**

zone *name* **getScavengeStartTime**

zone *name* **scavenge**

zone *name* **chkpt**

zone *name* **dumpchkpt** [**summary**]

Syntax Description

See Table 2-52 on page 2-199 for the **zone** command attributes and their descriptions.

The *name* is the fully qualified domain name (FQDN), including the trailing dot.

zone *name* **create primary file**=*hostfile* [*template*=*template-name*]

Creates a primary zone by importing data from the specified primary file and optionally applies the specified template:

```
nrcmd> zone example.com. create primary file=host.local
```

zone *name* **create primary** *nameserver person* [*attribute*=*valu* ...]

Creates a primary zone, along with the DNS nameserver and the person in charge (and optionally any additional attributes). Note that re-creating an existing zone overwrites the old one.

The **zone** command automatically creates the SOA and NS resource records for you. Use the **zone** *name* **addRR** command to create an A record for the nameserver that you specified in the *nameserver* value. This example creates an SOA record ns.test.org. andy.test.org. and an NS record ns.test.org:

```
nrcmd> zone test.org. create primary ns andy
```

Both of these records have the name of the zone ("test.org." or "@"). Because nameserver ns.test.org. is in the test.org. zone, you must also provide an A record for it:

```
nrcmd> zone test.org. addRR ns A 192.168.2.2
nrcmd> server dns reload
```

zone *name* **create secondary** *address* [*attribute*=*value* …]

Creates a secondary zone, along with the IP address of the primary nameserver for zone transfers (and optionally any additional attributes). The *address* becomes the *master-servers* attribute value (which can be an IP address-key combination).

zone *name* **applyTemplate** *template-name*

Applies a template to the zone, as created through the **zone-template** command.

zone *name* **delete**

Deletes a zone.

zone *name* **enable** *attribute*

Enables an attribute of a zone.

zone *name* **disable** *attribute*

Disables an attribute of a zone.

zone *name* **set** *attribute*=*value* [*attribute*=*value* ...]

Sets one or more attributes for a zone.

zone *name* **unset** *attribute*

Unsets the value of an attribute of the zone.

zone *name* **get** *attribute*

Gets the explicit value of an attribute of the zone.

zone *name* [**show**]

Shows the values of all attributes for a zone.

zone list

Lists all zones and their attributes.

zone listnames

Lists just the zone names.

zone *name* **forceXfer secondary**

Forces the secondary server to initiate a zone transfer:

```
nrcmd> zone test.org. forceXfer secondary
```

Note The **primary** argument is currently not implemented.

zone *name* **addHost** *hostname IPaddress* [*alias …*]

Adds the host name to a zone, along with its IP address and optional aliases. If a reverse zone exists that includes this host, this command also automatically creates a pointer (PTR) record in the reverse zone.

```
nrcmd> zone example.com. addHost bethpc 192.168.1.10
100 Ok
bethpc 192.168.1.10

nrcmd> zone 1.168.192.in-addr.arpa listRR
100 Ok
...
10   IN   PTR   bethpc.example.com
```

zone *name* **removeHost** *hostname*

Removes a host from a zone.

zone *name* **listHosts**

Lists the hosts for a zone.

zone *name* **addRR** [**staged | -sync**] *name* [*ttl*] [*class*] *type data*

Adds a protected resource record to a zone. The arguments for this command are in the same format as BIND files. You cannot add a protected record to an unprotected name.

zone *name* **removeRR** *name type data*

Removes specified protected resource records.

zone *name* **addDNSRR** *name* [*ttl*] *type data*

Adds an unprotected resource record to a zone. The name, type, and data must be specified. You cannot add an unprotected record to a protected name.

- *name*—Name of the resource record, which depends on its type. Required.
- *ttl*—Time-to-live of the resource record. If set to -1, then the TTL of the zone's SOA record applies.
- *type*—Type of resource record, for example PTR or A. For full descriptions, see the *Network Registrar User's Guide*. Required.
- *data*—Data that depends on the resource record type. Required.

For the resource record addition to take effect, you must reload the server. This example adds a Name Server (NS) resource record.

zone *name* **removeDNSRR** *name* [*ttl*] *type data*

Removes all unprotected resource records from a zone. Specify resource records by owner; owner and type; or owner, type, and data. Note that for the removal to take effect, you need to reload the server. See the attributes in the **addDNSRR** syntax description.

zone *name* [**protect-name | unprotect-name**] *name*

Sets the protection status of the resource records associated with the specified name.

zone *name* **syncToDns**

Synchronizes resource record data on the CCM server of the named zone to the DNS server.

zone *name* **listRR** {**all | ccm | dynamic**}

Displays the resource records for a zone. You can display all the resource records, or just the CCM or dynamic resource records.

zone *name* **getScavengeStartTime**

Gets the time for the next scheduled zone scavenging.

zone *name* **scavenge**

Causes scavenging on all zones that have enabled the *scvg-enabled* attribute.

zone *name* **chkpt**

Forces an update to the zone checkpoint database for the specified zone. Set the checkpoint interval using the **zone** *name* **set checkpoint-interval** command.

zone *name* **dumpchkpt** [**summary**]

Creates a human-readable file of the latest zone checkpoint. The optional summary keyword dumps a summary that includes the zone OID, zone checkpoint file name, zone checkpoint dump file name, resource record name-set count, and resource record count.

Attributes

Table 2-52 describes the **zone** command attributes.

Table 2-52 zone Command Attributes

Attribute	Usage	Description
checkpoint-interval	set get unset	Interval (in seconds) at which to checkpoint the zone (take the latest snapshot in the zone checkpoint database). Optional, no default.
defttl	set get unset	For a primary zone only, default TTL for this zone. Network Registrar responds to authoritative queries with an explicit TTL value, if one exists. If none exists, it responds with the default TTL value. Required for a primary zone, default 86400s (1d).
dist-map	set get unset	The zone distribution map associated with this zone. The zone distribution map describes the primary and secondary DNS servers that provide DNS service for this zone.
dynamic	**enable** disable	For a primary zone only, enables or disables RFC 2136 dynamic updates to the zone. The most typical source of these updates is a DHCP server. Required, default enable.

Table 2-52 zone Command Attributes (continued)

Attribute	Usage	Description
ixfr	enable disable unset	For a secondary zone only, enables or disables requesting incremental transfers. Overrides the ixfr-enable setting for the server. Optional, no default.
master-servers	set get unset	For secondary zones, the list of servers from which to transfer data. You can append each server address with an optional TSIG key name to configure secure zone transfers, in the syntax *address–key*. If you use the **zone** *name* **create secondary** *addr* command to create the secondary zone instead, the *addr* in that syntax becomes the *master-servers* value. Optional, no default; replaces the *auth-servers* attribute.
nameservers	set get unset	Comma-separated list of nameservers, in fully qualified domain name format. Optional, no default.
notify	enable disable unset	Enables notifying other authoritative servers when this zone changes. The setting overrides the global notify value for this zone. Optional, no default.
notify-set	set get	List of additional servers to notify when the zone changes. All servers listed in NS records for the zone, except the server described by the *ns* zone attribute (*mname* field of the SOA record), receive notifications. Network Registrar also notifies servers listed in the notify-set value. Optional, default empty.
origin	get	Fully qualified name of the zone's root. Read-only.
restrict-query-acl	set get	Limits client queries based on the source IP address, source network address, or ACL. The ACL can contain another ACL or a TSIG key. The ACL set on the DNS level *restrict-query-acl* serves as a filter for nonauthoritative queries. If a query is targeted at an authoritative zone, the zone's *restrict-query-acl* applies. However, if the query targets no authoritative zone, the DNS *restrict-query-acl* applies. Optional, default any.
restrict-xfer	enable **disable**	If enabled, restricts zone transfers to a specific set of hosts. If you restrict zone transfers, you need to use the restricted-set attribute to list the servers allowed to perform zone transfers. Required, default disable.
restrict-xfer-acl	set get	If *restrict-xfer* is enabled, the access control list (ACL) that designates who is allowed to receive zone transfers from this zone. Overrides the server setting. Optional, no default. Replaces the *restricted-set* attribute.
scvg-enabled	enable **disable**	For a primary zone only, enables or disables dynamic resource record scavenging of the zone. Use this for Microsoft clients, with other scavenging attribute settings. See the *Network Registrar User's Guide*. This setting overrides that on the server level. Required, default disable.
scvg-ignore-restart-interval	set get unset	For a primary zone only, the interval, in seconds, for which a server restart does not recalculate a start scavenging time. This setting overrides that on the server level. Optional, no default.
scvg-interval	set get unset	For a primary zone only, with **zone** *name* **enable scvg-enabled**, the interval, in seconds, at which the zone is scheduled for scavenging. This setting overrides that on the server level. Optional, server defaults apply.

Table 2-52 zone Command Attributes (continued)

Attribute	Usage	Description
scvg-no-refresh-interval	set get unset	For a primary zone only, with **zone** *name* **enable scvg-enabled**, the interval, in seconds, during which actions such as dynamic or prerequisite-only updates do not advance the timestamp for scavenging. This setting overrides that on the server level. Optional, server defaults apply.
scvg-refresh-interval	set get unset	For a primary zone only, with **zone** *name* **enable scvg-enabled**, the interval, in seconds, during which actions such as DNS updates and prerequisite-only updates advances the zone timestamp. This setting overrides that on the server level. Optional, server defaults apply.
subzone-forward	set get unset	For zones with forwarders set (through the **dns addForwarder** command), the normal Network Registrar behavior is to ignore delegation to subzone nameservers and forward queries to these forwarding servers instead. You would normally need to set a resolution exception (through the **dns addException** command) to the subzone server. This might be impractical for large numbers of subzones. With the *subzone-forward* attribute set to **no-forward**, when the server receives a query for any of its subzones, it tries to find relevant subzone NS records, resolve their corresponding IP addresses, and delegate the query to those IP addresses. Optional, default normal.
template	set get unset	Specifies the zone-template to apply to the zone. All properties configured on the zone-template are then applied to the zone.
update-acl	set get unset	Access control list (ACL) for allowing DNS updates to the zone. Use the **!** symbol for negation, for example: `nrcmd> zone example.com. set update-acl=acl1,!acl2` See the "acl" section on page 2-2 for the types of ACLs. Setting the attribute at the zone level overrides the server setting. (This attribute replaces the *dynupdate-set* attribute from the previous release, but has been since replaced by the **update-policy** command.) Optional, no default.
update-policy-list	set get unset	Named DNS update policy for the zone, as created by using the **update-policy** command (see the "update-policy" section on page 2-190). Optional. **Note** Be aware that if you set the update-acl attribute, then the update-policy-list might be ignored. If both update-acl and update-policy-list are unset on a zone, the zone defaults to using the dns server "update-acl" attribute if that is set. Network Registrar controls DDNS update authorization at the zone level with a single access control list and the implicit policy of not allowing updates to static resource records and only allowing updates to records of types A, TXT, PTR, CNAME and SRV.

Usage Guidelines

Importing Zone Data

To import an existing BIND zone file, in the CLI, create the zone using the **zone** *name* **create primary file**=*file* command. Reload the server after you import the file.

Network Registrar can read BIND named.boot and named.conf files and import all the zone files identified in them. Use UNIX file path syntax for all operating systems. Also, ensure that any $INCLUDE directives in zone files have absolute paths. Network Registrar makes any file paths relative to what any *directory* directive contains in the configuration file:

```
nrcmd> import named.boot /etc/named.boot
nrcmd> dns reload
```

Network Registrar recognizes $TTL directives for zone file imports. The first $TTL directive it encounters serves as the default TTL for the zone. This value is assigned to *defttl* for future use. Subsequent $TTL directives do not override the first directive; they do not change the default TTL for the zone. Instead, they provide the TTL for subsequent resource records that have no explicit TTL values. Consider this BIND zone file with $TTL directives:

```
$ORIGIN example.com.
@ IN SOA exampleDNSserv1 hostmaster 10 10800 3600 604800 7200
$TTL 3600
exampleDNSserv1 IN A 192.168.50.1
$TTL 7200
examplehost1 IN A 192.168.50.101
$TTL 9800
examplehost2 IN A 192.168.50.102
examplehost3 13400 IN A 192.168.50.103
```

Network Registrar imports this data as:

```
default TTL: 3600
example.com. IN SOA exampleDNSserv1 hostmaster 10 10800 3600 604800 7200
exampleDNSserv1 IN A 192.168.50.1
examplehost1 7200 IN A 192.168.50.101
examplehost2 9800 IN A 192.168.50.102
examplehost3 13400 IN A 192.168.50.103
```

Here is a sample named.conf file:

```
acl black-hats {
    10.0.2.0/24;
    192.168.0.0/24;
    };
acl red-hats {
    10.0.1.0/24;
    };
options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
    transfer-format many-answers;
    };
logging {
    category lame-servers { null; };
    };
key samplekey {
    algorithm hmac-md5;
    secret "c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
    };
controls {
    inet 127.0.0.1 allow { any; } keys { "key"; };
    };
zone "." in {
    type master;
    file "db.root";
    check-names ignore;
    allow-update { none; };
```

```
    allow-query { any; };
    allow-transfer { any; };
    notify no;
    };
zone "9.in-addr.arpa" in {
    type master;
    file "db.9";
    check-names ignore;
    allow-update { none; };
    allow-query { Any; };
    allow-transfer { 10.53.1.252;10.240.1.252;10.240.1.251; };
    notify yes;
    };
```

Creating a Primary Zone

Use the **zone** *name* **create primary** command to create a primary zone. (Note that you can import a zone file by using another command—see the "import" section on page 2-104.)

The minimum you have to specify to create a primary zone using the CLI is to give it a name, identify it as a primary zone, and add its primary DNS server and hostmaster (person in charge) names. The primary DNS server also becomes one of the authoritative nameservers for the zone. The CLI sets default values for all the other SOA record properties for the zone:

```
nrcmd> zone example.com. create primary exampleDNSserv1 hostmaster
```

This creates the example.com zone. The *exampleDNSserv1* entry is the name of the zone's primary DNS server. Note that it also appears as the first authoritative nameserver in the *nameservers* list. The *hostmaster* entry is the name of the person in charge of the zone, which appears as the value set for *person*. Enter this value in the syntax given at the beginning of this section.

The CLI sets the zone's serial number to 1 by default. If you want to change this setting, use the **zone** *name* **set serial** command. You must reload the server if you reset this number. Note that the DNS server does not necessarily recognize suggested serial number changes.

```
nrcmd> zone example.com. set serial=1
```

If you need to change the name of the zone's primary DNS server, use the **zone** *name* **set ns** command:

```
nrcmd> zone example.com. set ns=exampleDNSserv1
```

To change the hostmaster name, use the **zone** *name* **set person** command. Use the proper syntax:

```
nrcmd> zone example.com. set person=hostmaster.example.com.
```

To confirm the settings, use the **zone** *name* **get** command for each attribute, or the **zone list** or **zone** *name* **show** command to get all the attribute settings.

After you create the zone, reload the DNS server using the **dhcp reload** command:

```
nrcmd> dns reload
```

To delete the zone for any reason, use the **zone** *name* **delete** command and reload the server.

Creating a Secondary Zone

Use the **zone** *name* **create secondary** command. The IP address you include is that of the nameserver from which data is expected, typically a primary nameserver:

```
nrcmd> zone secondary.example.com. create secondary 192.168.50.1
```

To restrict zone transfers to particular addresses only, use the **zone** *name* **enable restrict-xfer** command, then use the **zone** *name* **set restrict-xfer-acl** command to specify the (comma-separated) addresses. Note that confirmation is given for the first address in the restricted set; use the **zone** *name* **show** command to display all addresses in the set:

```
nrcmd> zone secondary.example.com. enable restrict-xfer
nrcmd> zone secondary.example.com. set restrict-xfer-acl=192.168.1.1,192.168.1.20
nrcmd> zone secondary.example.com. show
```

Creating and Delegating Subzones

Use the **zone** *subzone* **create primary** and **zone** *subzone* **addRR** *hostname* **A** *address* commands to create the subzone and create an A record for the server. Then use the **zone** *parentzone* **addRR** *name* **NS** and **zone** *parentzone* **addRR** *hostname* **A** *address* commands to delegate the subzone on the parent zone. The last host record adds the glue record if the server is in the subzone:

```
nrcmd> zone boston.example.com. create primary bostonDNSserv1 hostmaster
nrcmd> zone boston.example.com. addRR bostonDNSserv1 A 192.168.60.1
nrcmd> zone example.com. addRR boston NS bostonDNSserv1.example.com.
nrcmd> zone example.com. addRR bostonDNSserv1 A 192.168.40.1
```

If you use the **zone** *name* **listRR** command for the parent zone, the NS and glue A records should appear:

```
nrcmd> zone example.com. listRR
```

To undelegate a subzone, use the **zone** *name* **removeRR NS** and **zone** *name* **removeRR A** commands to remove the subzone's NS and glue A records:

```
nrcmd> zone example.com. removeRR boston NS
nrcmd> zone example.com. removeRR bostonDNSserv1 A
```

To edit a subzone, use the **zone** *name* **removeRR** command to delete the NS and glue A records, use the **zone** *name* **addRR** command to replace them, then reload the DNS server:

```
nrcmd> zone example.com. removeRR boston NS
nrcmd> zone example.com. removeRR bostonDNSserv1 A
nrcmd> zone example.com. addRR boston NS bostonDNSserv2.example.com.
nrcmd> zone example.com. addRR bostonDNSserv2 A 192.168.40.2
nrcmd> dns reload
```

Exporting Zone Data

Exported BIND data can include static or dynamic addresses, or both. When exporting dynamic addresses sourced by a Network Registrar DHCP server, the data includes the host MAC addresses in a text (TXT) resource record for dynamically created records. To export a DNS zone, use the **export zone** command to specify the type of addresses (*static*, *dynamic*, or *both*) and the name of the output file. If you specify the filename without a path, the path defaults to the bin directory of the installation directories.

When Network Registrar receives an **export zone** CLI command, it records the default TTL for the zone in a BIND directive ($TTL). This example shows partial file output from an **export zone** command:

```
nrcmd> export zone example.com. static
100 Ok
$ORIGIN example.com.
$TTL 86400
@               IN      SOA     exampleDNSserv1.example.com. hostmaster.example.com. 2 10800
3600 604800 86400
@               IN      NS      exampleDNSserv1.example.com.
ns1             IN      CNAME   exampleDNSserv1.example.com.
exampleDNSserv1IN       A       192.168.50.1
examplehost1    IN      A       192.168.50.101
```

You can also export all zones of a particular type. Use the **export zonenames** {**forward** | **reverse** | **both**} *file* command to export the zone names to an output file.

Exporting UNIX Hosts Files

You can export DNS data in UNIX /etc/hosts file format. Network Registrar combines information from the A and CNAME records for a host. To export all the zones in the server in hosts file format, use the **export hostfile** command and give the name of the output file:

```
nrcmd> export hostfile
100 Ok
# Hostfile created by nrcmd from Network Registrar
# Cisco Systems, Inc.
# Created on Fri Jan 25 15:26:10 Eastern Daylight Time 2002
# 2 records created
192.168.50.1   exampleDNSserv1.example.com   exampleDNSserv1 ns1    #
192.168.50.101 examplehost1.example.com      examplehost1           #
```

Restoring a Loopback Zone

A loopback zone is a reverse zone whereby a host can resolve the loopback address (127.0.0.1) to the host name localhost so that it can direct traffic to itself. You could essentially omit the loopback zone without affecting the server, but a lookup of 127.0.0.1 would fail. Network Registrar normally creates a loopback zone for you, but you can configure it manually or import it from an existing BIND zone file.

If you accidentally delete the loopback zone, you can restore it without importing it:

Step 1 Use the **zone** *name* **create** command to create the loopback zone. This example adds the zone 127.in-addr.arpa, specifying that it is a primary zone, the nameserver is *localhost*, and the hostmaster is loopback:

```
nrcmd> zone 127.in-addr.arpa create primary localhost loopback
```

Step 2 Use the **zone** *name* **addRR** command to add the pointer (PTR) resource record.

Although the **zone** command automatically creates the NS and SOA record for you, you must use the **addRR** keyword to create a PTR record for the nameserver named in the *ns* field. This example adds the name 1.0.0, the type PTR, and the data localhost:

```
nrcmd> zone 127.in-addr.arpa addRR 1.0.0 PTR localhost
```

Step 3 Use the **zone** *name* **addRR** command to add the Address (A) resource record.

The A record provides the name-to-address mapping for the zone. This example adds the name localhost, the type A, and the data 127.0.0.1:

```
nrcmd> zone 127.in-addr.arpa addRR localhost A 127.0.0.1
```

Network Registrar automatically appends the zone name to the *ns*, *person*, and *data* fields; that is, localhost.127.in-addr.arpa and loopback.127.in-addr.arpa.

Step 4 Reload the DNS server.

You can also create the loopback zone using a BIND format loopback zone file and importing the file. This example adds the zone 127.in-addr.arpa primary zone and imports the BIND file, hosts.local:

```
nrcmd> zone 127.in-addr.arpa create primary file=hosts.local
nrcmd> dns reload
```

You can use this text to define the contents of the hosts.local file:

```
127.in-addr.arpa.43200 SOA localhost.127.in-addr.arpa.
loopback.127.in-addr.arpa. (
        1       ;serial
        3600    ;refresh
        3600    ;retry
        3600000 ;expire
        43200 ) ;minim
127.in-addr.arpa.                       IN      NS      localhost.127.in-addr.arpa.
1.0.0.127.in-addr.arpa.         86400   IN      PTR     localhost.127.in-addr.arpa.
localhost.127.in-addr.arpa.     86400   IN      A       127.0.0.1
```

Related Commands **acl**, **dns**, **server**, **update-policy**, **zone-dist**, **zone-template**

zone-dist

The **zone-dist** command enables you to define and manage zone distribution configurations.

On local clusters, use the **zone-dist sync** command to synchronize staged edits to the DNS server and to synchronize primary zones to secondary zones. Regardless of the mode selected, the exact list of authoritative zones (primary and secondary) will be synchronized with the DNS server.

On the regional cluster, use the **zone-dist sync** command to synchronize primary zones from the regional configuration to the primary local cluster and to synchronize primary zones to secondary zones. Primary zones on the local cluster are placed 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 master servers list specified by the distribution map. In exact mode, any zones not matching the distribution map are deleted.

zone-dist *name* create *primary-cluster* [*attribute*=*value* ...]

zone-dist *name* delete

zone-dist list

zone-dist listnames

zone-dist *name* **show**

zone-dist *name* **get** *attribute*

zone-dist *name* **set** *attribute*=*value*...

zone-dist *name* **unset** *attribute*

zone-dist *name* **addSecondary** *secondary-cluster* [*master-server-ipkeys*]

zone-dist *name* **removeSecondary** *secondary-cluster* [*master-server-ipkeys*]

zone-dist *name* **listSecondaries**

zone-dist *name* **sync** [**update** | **complete** | **exact**] [**no-rrs**] [**no-secondaries**]

Attributes

Table 2-53 describes the **zone-dist** command attributes and their values and defaults, if any.

Table 2-53 zone-dist Command Attributes

Attribute	Usage	Description
master-servers	get set unset	The list of master-servers to set when creating a secondary zone on secondary server. Optional, no default. Valid values are one or more IP addresses with optional keynames: *ipaddress*[**-***keyname*].
name	get	The name of this zone distribution map. Required, set at creation.
primary	get set unset	The cluster or HA DNS pair serving the primary zones associated with this zone distribution map. Optional, no default.

Usage Guidelines

The [no-rrs] and [no-secondaries] flags can be used to skip portions of the synchronization logic. This can improve the performance of the command, but should only be used when you are certain there are no changes pending. For example, if primary zones are up-to-date with the DNS server, you can use the [no-rrs] flag to synchronize your secondary zones.

zone-template

The **zone-template** command configures DHCP failover server pairs.

zone-template *name* **create**

zone-template *name* **apply-to** [**all** | *zone1*,*zone2*,...]

zone-template *name* **delete**

zone-template *name* **set** *attribute*=*value* [*attribute*=*value*…]

zone-template *name* **get** *attribute*

zone-template *name* [**show**]

zone-template list

Syntax Description

zone-template *name* **apply-to** [**all** | *zone1*,*zone2*,...]

Applies a zone template to one or more zones.

Attributes

Table 2-52 on page 2-199 describes the **zone-template** command attributes and their values and defaults, if any.