



## Managing IPv6 Addresses

Network Registrar 6.2 now supports IPv6 addressing for DHCP (DHCPv6) and supports:

- Stateless autoconfiguration (RFC 3736)—The DHCPv6 server does not assign addresses, but instead provides configuration parameters, such as DNS server data, to clients.
- Stateful autoconfiguration (RFC 3315)—The DHCPv6 server assigns nontemporary or temporary addresses and provides configuration parameters to clients.
- Prefix Delegation (RFC 3633)—The DHCPv6 server does prefix delegation to clients (routers).

The DHCPv6 service provides these basic capabilities:

- Links and prefixes—Similar to DHCPv4 networks and scopes that define the network topology. Each link can have one or more prefixes.
- Policies and options—Attributes and options can be assigned to links, prefixes, and clients.
- VPN support—Provides multiple address spaces.
- Client classing—Clients can be classified and prefixes can be selected based on known clients or packet-based expressions.
- Static reservations—Clients can receive predetermined addresses.
- Statistics collection and logging—Provides server activity monitoring.

The DHCPv6 service requires that the server's operating system supports IPv6 and that at least one interface on the system is configured for IPv6. An IPv6 network infrastructure with DHCPv6 clients (and DHCPv6 relay agents) are required for successful operation.

You will also need a Network Registrar IPv6 license key.

## DHCPv6 Concepts

The following sections describe the concepts related to DHCPv6 operation.

### IPv6 Addressing

IPv6 addresses are 128 bits long and are represented as a series of 16-bit hexadecimal fields separated by colons (:). The A, B, C, D, E, and F in hexadecimal are case insensitive. For example:

```
2031:0000:130f:0000:0000:09c0:876a:130b
```

A few shortcuts to this addressing are:

- Leading zeros in a field are optional, so that 09c0 can be written 9c0, and 0000 as 0.
- Successive fields of zeros (any number of them) are represented as ::, but only once in an address (because if used more than once, the address parser has no way of identifying the size of each block of zeros). So, the previous address can be written:

```
2031:0:130f::09c0:876a:130b
```

The use of the double-colon abbreviation makes many addresses small, for example ff01:0:0:0:0:1 becomes **ff01::1**.

Link-local addresses have a scope limited to the link, and use the prefix fe80::/10. Loopback addresses have the address ::1. Multicast addresses are identified by the prefix ff00::/8 (there are no broadcast addresses in IPv6).

The IPv4-compatible addresses in IPv6 are the IPv4 decimal quad addresses prefixed by ::. For example, an IPv4 address that would be interpreted as ::c0a8:1e01 can be written as ::**192.168.30.1**.

## Links and Prefixes

The explicit DHCPv6 configuration objects are links and prefixes:

- **Link**—Network segment that can have one or more prefixes, and adds an additional layer at which policies can be applied for DHCPv6 clients.
- **Prefix**—Equates to a scope in IPv4. The link associated with a prefix is similar to a scope's primary-scope, except that it names a link and not another prefix.

Just as with scopes, multiple prefix objects for the same IPv6 prefix may be created. However, rather than supporting multiple ranges with explicit start and end addresses, prefixes only support a single range that must be an IPv6 prefix with a length the same as, or longer than, that of the prefix. If a range is not specified, the IPv6 prefix is used as the range. A range is limited to powers of 2. The range must be unique (it cannot be duplicated by any other range, except in a different VPN) and cannot be contained in, or contain, another range.

A link needs to be created only if more than one prefix object with a different IPv6 prefix exists on a link. When the server loads the configuration, if a prefix has no explicit link, the server searches for or creates an implicit link with the name Link-[*vpn.name*]/*prefix*. All prefix objects with the same IPv6 prefix must either not specify a link or explicitly specify the same link.

The DHCPv6-enabled server supports VPNs (namespaces) for DHCPv6. However there is presently no means to make use of anything other than the default global VPN (there is no VPN option). Both the link and prefix objects have a *vpn-id* attribute, since prefixes do not require links, but all prefixes on a link must use the same *vpn-id*.

## Determining Links and Prefixes

When the DHCPv6 server receives a DHCPv6 message, it determines the link and prefixes to be used to service the request as follows:

1. Finds the source address:
  - a. If the client message was relayed, the server sets the source address to the first nonzero link-address field starting with the Relay-Forward message closest to the client (working outwards). If the server finds a source address, it proceeds to step 2.
  - b. Otherwise, if the message source address is a link-local address, the server sets the source address to the first address for the interface on which the message was received for which a prefix exists (or 0 if no prefix is found for any address). It then proceeds to step 2.
  - c. Otherwise, the server sets the source address to the message source address.
2. Locates the prefix for the source address. If the server cannot find a prefix for the source address, it cannot service the client and drops the request.
3. Locates the link for the prefix. This always exists and is either an explicitly configured link or the implicitly created link based on the prefix address.

Now that the server can determine the client's link, it can process the client's request. Depending on whether the client's request is stateful or prefix-delegated, and on the selection criteria and other factors, one or more prefixes for the link might be used to service the client's request.

This is one area of difference between DHCPv4 and DHCPv6. In DHCPv4, only one of the scopes from the network is selected to service the client's request. In DHCPv6, all of the prefixes for the link are potentially usable. Thus, a client might be assigned an address, or delegated a prefix, from multiple prefixes for the link (subject to selection criteria and other conditions).

## Generating Addresses

IPv6 addresses are 128-bit addresses (as compared to 32-bit addresses for IPv4). In most cases, DHCPv6 servers assign at least 64 of those bits. Because of this, addresses are generated using the client's 64-bit interface-identifier whenever possible, or through a random number generator. The interface-identifier emulates how stateless autoconfiguration assigns addresses to the client. Unfortunately, there are privacy concerns regarding autoconfiguration, and it does not work if the client requests multiple addresses on the same prefix. The alternative is the random number generator (see RFC 3041). You can configure addresses whether or not the interface-identifier is assigned first, if available.

## Generating Delegated Prefixes

The DHCPv6 server uses a first-fit algorithm when generating delegated prefixes. Depending on the configured default, minimum, and maximum, and the client's requested prefix length, it returns the first available shortest prefix to the client.

## DHCPv6 Clients and Leases

The DHCPv6 server supports clients and leases that are similar to those for DHCPv4. The key differences are:

- DHCPv6 clients are identified by their DHCP Unique Identifier (DUID), which is the DHCPv4 concept of hardware addresses and client IDs consolidated into one unique client identifier.
- DHCPv6 clients can have multiple leases. This means that if multiple prefixes are on a single link, the client is assigned an address from each prefix that it is allowed to use, not just from one scope, as in DHCPv4.
- A DHCPv6 client is created when the first lease is associated with it and deleted when it no longer has any leases associated with it. This is identical to DHCPv4 behavior, except that a DHCPv4 client can only have a single lease.
- DHCPv6 leases are dynamically created. All leases that the server can potentially use are not created at configuration time, because there could potentially be billions of these leases.

Leases can be for:

- Nontemporary addresses—Standard IPv6 unicast addresses with likely long (and renewable) lifetimes.
- Temporary addresses—Standard IPv6 unicast addresses, but with very limited (and nonrenewable) lifetimes. Temporary addresses solve a privacy issue with IPv6 (see RFC 3041).
- Delegated prefixes—Used for prefix delegation (see RFC 3633).

Leases have both a preferred and valid lifetime:

- Preferred lifetime—Primarily for the client's use, the length of time that a valid address is preferred. When the preferred lifetime expires, the address becomes deprecated.
- Valid lifetime—Used by both client and server, it is the length of time an address remains in the valid state. The valid lifetime must be greater than or equal to the preferred lifetime. When the valid lifetime expires, the address becomes invalid. A lease is eligible to be deleted once the valid lifetime expires. This is essentially the same as the DHCPv4 lease time.

## Lease Affinity

In DHCPv4, when a lease expires or is released, the server remembers the client for an address as long as it is not assigned to another client. DHCPv6 does not do this. During the affinity period, the lease is in the AVAILABLE state and still associated with the client that last leased it. If the client requests a lease during this affinity period, it is granted the same lease (or, if renewals are inhibited, it is explicitly not given that lease). Because of the large IPv6 address space and depending on the address generation technique, it could be millions of years before an address ever needs to be reassigned to a different client, and there is no reason to have to hold on to this information for that long.

Because of this, Network Registrar provides an *affinity-period* attribute so that the client can get the same address even if not requesting a renewal before expiration. This is desirable in some environments, but not in others where the affinity time would be zero or very small.

## Lease Life Cycle

Leases have a life cycle controlled by states. A lease only exists while it is associated with a client and is deleted once it is no longer associated with that client. The life cycle and state transitions are:

1. A lease is born and associated with an address when:
  - a. A reservation is created for a lease—This puts the lease in the `AVAILABLE` state and marks it as `RESERVED`. No timer is associated with this state and the lease is not deleted as long as it is `RESERVED`.
  - b. An `ADVERTISE` message is sent to a client—This puts the lease in `OFFERED` state. It will transition to `DELETED` state after the offer-timeout.
  - c. A `REPLY` message is sent to a client (for a `REQUEST`, `RENEW`, or `REBIND`)—This puts the lease in `LEASED` state. It will transition to `EXPIRED` state after the valid lifetime for the lease elapses.
2. An `OFFERED` lease transitions to:
  - a. `LEASED` state when a `REQUEST` message is received, and then transitions to `EXPIRED` state after the valid lifetime for the lease elapses.
  - b. `DELETED` state if the offered-time expires.
3. A `LEASED` lease:
  - a. Is renewed when a `REQUEST`, `RENEW`, or `REBIND` message is received. It transitions to `EXPIRED` state after the new valid lifetime for the lease elapses (note that the new valid lifetime could be 0).
  - b. Transitions to `RELEASED` state when a `RELEASE` message is received. It transitions to `AVAILABLE` state after the release-grace-period elapses.
  - c. Transitions to `UNAVAILABLE` state when a `DECLINE` message is received. It is deleted after the unavailable-timeout period elapses.
4. An `EXPIRED` lease transitions to `AVAILABLE` state after the grace-period. It is deleted after the affinity-period elapses.
5. An `AVAILABLE` lease:
  - a. Transitions to `DELETED` state and is deleted from memory and the lease database after the affinity-period elapses.
  - b. Cannot be deleted if it is `RESERVED`, and it remains `AVAILABLE`.
6. A `LEASED`, `EXPIRED`, `RELEASED`, or `AVAILABLE` lease can be reoffered to a client but remains in its current state, although the timeout is extended to at least the *offer-timeout*.
7. A `LEASED` lease can also transition to `REVOKED` state if the server needs to revoke the lease (the lease transitions to `AVAILABLE` state after the valid lifetime expires). A leased lease may be revoked when the client attempts to renew, if the lease is reserved for a different client or the prefix is no longer usable.

## DHCPv6 Reservations

Reservations are allowed only for nontemporary addresses and delegated prefixes. They are stored under the prefix in the configuration and must always be for an address (or prefix) under the prefix. The reservation can be outside the prefix object's range, provided that it is not within another prefix object's range. This has implications when adding new prefix objects, because existing reservations could violate this rule.

## DHCPv6 Bindings

Bindings are new to DHCPv6 and allow multiple groups of addresses to be allocated to a client. A client's binding consists of one of three types:

- Non-temporary (IA\_NA)
- Temporary (IA\_TA)
- Prefix delegation (IA\_PD)

A binding also consists of a unique Identity Association Identifier (IAID). Leases always exist under a binding. Clients, therefore, have one or more bindings, and bindings have one or more leases. Bindings are created when the first lease is added and removed when it has no more leases. Clients are created when the first binding is added, and removed when it has no more bindings.

## DHCPv6 Policy Hierarchy

DHCPv6 uses the existing policy objects, with additional DHCPv6 specific attributes (that are mostly analogous to those in DHCPv4). For DHCPv6, the hierarchy is:

1. Client embedded policy.
2. Client named policy
3. Client-class embedded policy
4. Client-class named policy
5. Prefix embedded policy
6. Prefix named policy
7. Link embedded policy
8. Link named policy
9. system\_default\_policy

This hierarchy is the same as for DHCPv4, except for the additional link policies and the fact that the prefix policies replace the scope policies. (For a comparison with the DHCPv4 policy hierarchy, see the [“Options Reply Processing” section on page 20-2.](#))

The hierarchy applies to most policy attributes, which are processed in the context of a single prefix. However, a few attributes (specifically *allow-rapid-commit*, *v6-reply-option*, *v6-options*, and *v6-vendor-options*) are processed in the context of multiple prefixes. In these cases, the processing at the prefix levels (steps 5 and 6) is a bit different:

- If the client requests Rapid Commit (see the [“Editing DHCPv6 Server Attributes” section on page 25-10](#)), the DHCP server checks the embedded and named policies of all the active prefixes on the link that the client is allowed to use (based on selection tags). If one of these policies has *allow-rapid-commit* disabled, the client's request is processed as if Rapid Commit were not part of the request. If at least one policy has *allow-rapid-commit* enabled, then the client can use Rapid Commit. If no prefix policy has the attribute configured, then processing continues at step 7.
- For the options-related attributes (see the [“Setting DHCPv6 Options” section on page 25-12](#)), the server also does special handling at steps 5 and 6. The server checks the embedded and then named policy of each active prefix on the link. It then uses the first one with the configured *v6-reply-option* attribute, or the first one with the configured value for the *v6-options* or *v6-vendor-options*.
- The prefixes are checked in case-insensitive alphabetical order.

**Tip**

In configurations with multiple prefixes on a link, avoid setting the Rapid Commit and option properties for the prefix policy, but rather set them on the link policy or other policy instead.

## DHCPv6 Options

DHCPv6 options do not use any DHCPv4 options; they are unique and separate. At present, there are about 30 DHCPv6 options. Most of these options are the DHCPv6 protocol infrastructure options and are not user definable. They use a 16-bit option code and 16-bit length (DHCPv4 uses only 8 bits for both of these). The configuration of options and behavior of configured options in policies are similar to those for DHCPv4. See the “[Setting DHCPv6 Options](#)” section on page 25-12 for details about client processing as it relates to the policy hierarchy.

## Supported and Unsupported Features

Features supported and not supported in this release are:

- LDAP with DHCPv6 enabled is supported.
- DHCP extensions are not supported.
- DNS updates are not supported.

## DHCPv6 Configuration

The following sections describe how to configure DHCPv6 in Network Registrar.

**Note**

You need a Network Registrar IPv6 license key to access the DHCPv6 features. Without this key, the IPv6-specific menu picks are not displayed in the Web UI.

## Configuring Links

- Step 1** In the Web UI, click **DHCP**, then **Links**. The List DHCPv6 Links page shows the existing links. (If you do not see the Links function, you do not have the IPv6 license installed.)
- Step 2** To add a link, click **Add Link**.
- In the CLI, use **dhcp-link name create** For example:
- ```
nrcmd> dhcp-link example-link create
```
- Step 3** In the Web UI, on the Add DHCPv6 Link page (see [Figure 25-1](#)), enter at least the text name you wish to give the link. You can also specify a policy and a VPN for the link.

Figure 25-1 Add DHCPv6 Link Page (Local)

| Attribute   | Value                | Data Type | Default |
|-------------|----------------------|-----------|---------|
| name*       | <input type="text"/> | string    |         |
| description | <input type="text"/> | string    |         |
| policy      | [none] ▼             | policy    | default |
| vpn-id      | [none] ▼             | VPN       |         |

149423

**Step 4** Click **Add Link**.

In the CLI, you can set and enable attributes in the usual way, and you can show and list links.

## Configuring Prefixes

**Step 1** In the Web UI, click **DHCP**, then **Prefixes**. The List DHCPv6 Prefixes page shows the existing prefixes. (If you do not see the Prefixes function, you do not have the IPv6 license installed.)

**Step 2** To add a prefix, click **Add Prefix**.

In the CLI, use **dhcp-prefix name create ipv6address/length**. For example:

```
nrcmd> dhcp-prefix example-prefix create 3ffe:8000::/64
```

**Step 3** In the Web UI, on the Add DHCPv6 Prefix page (see [Figure 25-2](#)), enter at least the text name you wish to give the prefix and the IPv6 address value with its prefix length. You can also choose or enter optional values, such as the associated VPN, whether the prefix should be assigned to stateful or prefix-delegation, and any associated policies or links.

Figure 25-2 Add DHCPv6 Prefix Page (Local)

| Attribute                   | Value                                                        | Data Type                 | Default  |
|-----------------------------|--------------------------------------------------------------|---------------------------|----------|
| name*                       | <input type="text"/>                                         | string                    |          |
| vpn-id                      | [none] v                                                     | VPN                       |          |
| address*                    | <input type="text"/> 64 v                                    | IPv6 prefix/prefix length |          |
| dhcp-type                   | [dhcp] v                                                     | 32-bit enum               | dhcp     |
| range                       | <input type="text"/> 64 v                                    | IPv6 prefix/prefix length |          |
| prefer-interface-identifier | <input type="radio"/> true <input type="radio"/> false       | boolean                   | false    |
| expiration-time             | <input type="text"/>                                         | date                      |          |
| policy                      | [none] v                                                     | policy                    |          |
| selection-tags              | <input type="text"/>                                         | list                      |          |
| link                        | [none] v                                                     | link                      |          |
| deactivated                 | <input type="radio"/> enabled <input type="radio"/> disabled | boolean                   | disabled |
| ignore-declines             | <input type="radio"/> true <input type="radio"/> false       | boolean                   | false    |

| Reservations             | IPv6 Address         | Lookup Key           |                                                              |
|--------------------------|----------------------|----------------------|--------------------------------------------------------------|
| <input type="checkbox"/> | <input type="text"/> | <input type="text"/> | <input type="radio"/> string<br><input type="radio"/> binary |
| Add Reservation          |                      |                      |                                                              |
| Add Prefix               |                      | Cancel               |                                                              |

149422

- Step 4** If there any lease reservations assigned to the prefix, give each an IPv6 address and lookup key, then click **Add Reservation**.
- Step 5** Click **Add Prefix**.
- Step 6** To edit a prefix, click the name of the prefix on the List/Add DHCPv6 Prefix page. On this page, you can edit the prefix properties, and also create a new embedded policy:
- Click **Create New Embedded Policy** to open the Edit DHCP Embedded Policy for Prefix page.
  - Modify the embedded policy properties (see the “[DHCPv6 Policy Hierarchy](#)” section on page 25-6).
  - Click **Modify Embedded Policy**. The next time the Edit DHCPv6 Prefix page appears, you can edit the embedded policy for the prefix.
  - Complete editing the prefix by clicking **Modify Prefix**.
- Step 7** Once you create a prefix, you can list the leases for the prefix by clicking the View icon (63) in the Leases column of the List DHCPv6 Prefixes page. This opens the List DHCP Leases for Prefix page. Click **Return** after viewing the leases.

In the CLI, you can set and enable attributes in the usual way. Add reservations using **dhcp-prefix name addReservation ipv6address/length lookup-key [-blob | -string]**. List leases using **dhcp-prefix name listLeases**. Manage DHCPv6 leases by using these commands:

```
nrcmd> lease6 {vpn-id/ | vpn-name/} ip6address[/prefix-length] activate
nrcmd> lease6 {vpn-id/ | vpn-name/} ip6address[/prefix-length] deactivate
nrcmd> lease6 {vpn-id/ | vpn-name/} ip6address[/prefix-length] force-available
nrcmd> lease6 {vpn-id/ | vpn-name/} ip6address[/prefix-length] get attribute
nrcmd> lease6 {vpn-id/ | vpn-name/} ip6address[/prefix-length] show
nrcmd> lease6 list
```

## Editing DHCPv6 Server Attributes

- 
- Step 1** In the local Web UI, click **DHCP**, then **DHCP Server** to open the Manage DHCP Server page.
- Step 2** Click the Local DHCP Server link to open the Edit DHCP Server page.
- Step 3** The DHCPv6 attributes to modify are:
- *v6-client-class-lookup*—Expression that determines a client-class based on the DHCPv6 client request and returns a string or <none>.
  - *max-client-leases*—Maximum number of leases a DHCPv6 client can have on a link (default 200).
- In the local cluster CLI, use the **dhcp** command to show these DHCP server attributes, then modify them using **dhcp set**.
- Step 4** Click **Modify Server**.
- 

## Configuring DHCPv6 Policies

- 
- Step 1** In the Web UI, click **DHCP**, then **Policies** to open the List DHCP Policies page.
- Step 2** Click an existing policy to open the Edit DHCP Policy page, or click **Add Policy** to add a new policy on the Add DHCP Policy page. Both pages have a DHCPv4 Options and a DHCPv6 Options section.
- Policies have these DHCPv6 attributes:
- *allow-rapid-commit*—With Rapid Commit enabled, clients receive information (when solicited) on committed addresses, which are then more quickly committed with a client request (default disable). Use Rapid Commit only if one DHCP server is servicing clients, otherwise it might seem like the client is receiving multiple addresses. (See the “[DHCPv6 Policy Hierarchy](#)” section on page 25-6 for special handling of this attribute when used in an embedded or named policy for a prefix.)
  - *allow-client-hints*—Enable or disable using addresses, prefixes, and lifetime that DHCPv6 clients request in client messages (default disable).
  - *allow-non-temporary-addresses*—Enable or disable DHCPv6 clients requesting nontemporary (IA\_NA) addresses (default enable).
  - *allow-temporary-addresses*—Enable or disable DHCPv6 clients requesting temporary (IA\_IA) addresses (default enable).
  - *affinity-period*—See the “[Lease Affinity](#)” section on page 25-4 (no default).
  - *preferred-lifetime*—Default and maximum preferred lifetime for leases (default 1 week).
  - *valid-lifetime*—Default and maximum valid lifetime for leases (default 2 weeks).
  - *v6-reply-options*—DHCPv6 options returned in replies to clients (no default). (See the “[DHCPv6 Policy Hierarchy](#)” section on page 25-6 for special handling of this attribute when used in an embedded or named policy for a prefix.)
  - *default-prefix-length*—For prefix delegation, default prefix length of the delegated prefix if the client or router does not explicitly request it (or *allow-client-hints* is disabled); must always be less than or equal to the prefix range’s prefix length (default 64 bytes).
  - *longest-prefix-length*—For prefix delegation, longest prefix length of the delegated prefix. If the client or router requests a longer prefix, this value is used (default is the default prefix length).

- *shortest-prefix-length*—For prefix delegation, shortest prefix length of the delegated prefix. If the client or router requests a shorter prefix, this value is used (default is the default prefix length).

In the local cluster CLI, use **policy list** to show these policy attributes, then modify them using **policy name set** or **enable**.

**Step 3** Click **Modify Policy**.

---

## Configuring DHCPv6 Client-Classes

**Step 1** In the Web UI, click **DHCP**, then **Client-Classes** to open the List DHCP Client-Classes page.

**Step 2** Click an existing client-class to open the Edit DHCP Client-Class page, or click **Add Client-Class** to add a new client-class on the Add DHCP Client-Class page. Both pages include these attributes:

- *v6-client-lookup-id*—Key value to use to look up the DHCPv6 client in the client database (locally or through LDAP), specified as an expression that evaluates to a string (or a blob as a valid string).
- *v6-override-client-id*—Value that replaces any client-identity value in an incoming packet, specified as an expression that evaluates to a blob.

In the local cluster CLI, use **client-class list** to show these client-class attributes, then modify them using **client-class name set**.

**Step 3** Click **Modify Client-Class**.

**Step 4** To generate clients, be sure that *validate-client-name-as-mac* is disabled for the DHCP server. In the Web UI, this attribute appears on the Edit DHCP Server page under the Client-Class attributes.

**Step 5** Reload the DHCP server.

---

## Configuring DHCPv6 Clients

**Step 1** In the Web UI, click **DHCP**, then **Clients** to open the List/Add DHCP Clients page.

**Step 2** Click an existing client to open the Edit DHCP Client page, or click **Add Client** to add a new client-class on the List/Add DHCP Client page.

In the local cluster CLI, use **client list** to show the existing clients.

**Step 3** Choose the client-class that includes the DHCPv6 attributes that were set (see the “[Configuring DHCPv6 Client-Classes](#)” section on page 25-11).

In the CLI, use **client name set client-class-name=value**.

**Tip** Also be sure that the *validate-client-name-as-mac* attribute is disabled for the DHCP server.

**Step 4** Click **Modify Client**.

---

## Setting DHCPv6 Options

Set DHCPv6 options and vendor options when you create or edit policies (embedded or named) for prefixes. In the Web UI, the DHCPv6 options coexist along with the DHCPv4 options on the Add DHCP Policy or Edit DHCP Policy page. Note that the vendor options appear only if you create these options (see the [“Creating DHCP Option Definition Sets and Option Definitions”](#) section on page 20-6).

You can select the options from the drop-down lists. If option descriptions exist, they appear under the Name and Number headings, which you can click to sort the entries.

In the CLI, use **policy name setV6Option** or **policy name setV6VendorOption**. The option settings require an option name (or ID) and a value. For example:

```
nrcmd> policy dhcpv6-policy setV6Option dns-servers 2222::1,2222::2
nrcmd> policy dhcpv6-policy setV6VendorOption 1234 2222::3,2222::4
```

(See the [“DHCPv6 Policy Hierarchy”](#) section on page 25-6 for special handling of the *v6-options* and *v6-vendor-options* policy attributes when used in an embedded or named policy on a prefix.)