



Managing DHCPv6 Addresses

Cisco Prime Network Registrar supports the following IPv6 addressing for DHCP (DHCPv6):

- **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 delegates prefixes to clients (routers).

The DHCPv6 service provides these capabilities:

- **Allocation groups**—Allows multiple prefixes to be treated as one from an allocation standpoint, and provides control over the order in which the prefixes are used.
- **Client-classing**—You can classify clients and select prefixes based on known clients or packet-based expressions.
- **DNS Updates**—DNS server updates of DHCP activity (over IPv4).
- **Extensions**—Extend the DHCP server processing by using C/C++ and Tcl extensions.
- **Failover**—You can configure a DHCP failover pair so that if one cannot provide leases to requesting clients, another one can take over.
- **LDAP**—Allows client entry lookups in an LDAP repository (external to Cisco Prime Network Registrar) and these clients may specify client reservations. LDAP writes/updates are not supported for DHCPv6.
- **Leasequery**—Offers leasequery support.
- **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**—You can assign attributes and options to links, prefixes, and clients.
- **Prefix Stability**—Clients can retain the delegated prefix when they change their location, that is even when they move from one CMTS to another or move within an address space. Prefix Stability, with appropriate infrastructure support (CMTS, routers), allows the subscriber to be moved or move without requiring a different delegated prefix.
- **SNMP traps**—Generate traps for events, such as if the number of leases in a prefix exceeds a certain limit (or drops below a certain limit) or if the server detects duplicate addresses.
- **Reservations**—Clients can receive predetermined addresses.
- **Statistics collection and logging**—Provides server activity monitoring.
- **VPN support**—Provides multiple address spaces (virtual private networks).

The DHCPv6 service requires that the server operating system support IPv6 and that you configure at least one interface on the system for IPv6.

Related Topics

- [DHCPv6 Concepts, page 28-2](#)
- [DHCPv6 Configuration, page 28-13](#)
- [DNS Update for DHCPv6, page 28-36](#)

DHCPv6 Concepts

The following subsections describe the concepts related to DHCPv6 operation:

- [IPv6 Addressing](#)
- [Links and Prefixes](#)
- [DHCPv6 Clients and Leases, page 28-7](#)
- [DHCPv6 Policy Hierarchy, page 28-12](#)
- [DHCPv6 Options, page 28-13](#)

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:

```
2001:db8:0000:0000:0000:0000:0000:0000
```

A few shortcuts to this addressing are:

- Leading zeros in a field are optional, so that you can write 09c0 as 9c0, and 0000 as 0.
- You can represent successive fields of zeros (any number of them) by a double colon (::), 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). This reduces the length of addresses; for example, 2001:db8:0000:0000:0000:0000:0000:0000 can be written:

```
2001:db8:::
```

Link-local addresses have a scope limited to the link, and use the prefix fe80::/10. Loopback addresses have the address ::1. Multicast addresses have 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, you can write the IPv4 address interpreted as **::c0a8:1e01** in the form **::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 primary scope, except that it names a link and not another prefix.

Just as with scopes, you can create multiple prefix objects for the same IPv6 prefix. However, rather than supporting multiple ranges with explicit start and end addresses, prefixes support only a single range that must be an IPv6 prefix with a length the same as, or longer than, the prefix object. For example, if you define a 2001::/64 prefix with a 2001::/96 range, the server can assign addresses from 2001:0:0:0:0:0:0:0 through 2001:0:0:0:0:0:ffff:ffff only. The range:

- Is limited to powers of 2.
- Must be unique (cannot be duplicated by any other range, except in a different VPN).
- Cannot be contained in, or contain, another range, except for prefix delegation prefixes, as explained below.
- Is the full IPv6 prefix if not specified, except for prefix delegation prefixes, as explained below.

If a prefix delegation prefix object is defined with an unspecified range, it may contain non prefix-delegation prefixes, and the effective range is either:

- The full IPv6 prefix if no other prefixes exist with the same IPv6 prefix, or
- The prefixes that remain when all other ranges for prefix objects with the same IPv6 prefix are removed from the IPv6 prefix.

You create a link 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 VPN address spaces for DHCPv6. Both the link and prefix objects may be assigned to a VPN. But all prefixes on a link must use the same VPN ID. Because there is presently no DHCPv6 VPN option, clients can only be assigned addresses from a VPN by using the client or client-class *override-vpn* attribute.

Related Topics

- [Determining Links and Prefixes](#)
- [Generating Addresses, page 28-4](#)
- [Generating Delegated Prefixes, page 28-5](#)
- [Prefix Stability, page 28-5](#)
- [Prefix Allocation Groups, page 28-6](#)

Determining Links and Prefixes

When the DHCPv6 server receives a DHCPv6 message, it determines the links and prefixes it uses to service the request. The server:

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 it received the message for which a prefix exists (or 0 if it finds no prefix 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.

- 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. The link must be a topological link (see the “[Prefix Stability](#)” section on page 28-5 section below).

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

This is one area of difference between DHCPv4 and DHCPv6. In DHCPv4, the server selects only one of the scopes from the network to service the client request. In DHCPv6, the server can use all the prefixes for the link. Thus, the server might assign a client an address, or delegate a prefix, from multiple prefixes for the link (subject to selection criteria and other conditions). (See the “[Configuring Links](#)” section on page 28-14.)

Generating Addresses

IPv6 addresses are 128-bit addresses (as compared to 32-bit addresses for IPv4). In most cases, DHCPv6 servers assign 64 of those bits, the interface-identifier (EUI-64) portion (see RFC 4291). You can generate addresses by using the client 64-bit interface-identifier or a random number generator. The interface-identifier emulates how stateless autoconfiguration assigns addresses to clients. Unfortunately, there are privacy concerns regarding its use, and it is limited to one address per prefix for the client.

By default, Cisco Prime Network Registrar generates an address using an algorithm similar to that described in RFC 4941 to generate a random interface identifier. These random interface identifiers have a zero value for the universal/local bit to distinguish them from EUI-64-based identifiers. The server also skips randomly generated interface identifiers from `::0` to `::ff` so that you can use identifiers for infrastructure devices (such as routers). You can configure whether to assign the interface-identifier (if available) first for each prefix (through the interface-identifier flag of the prefix *allocation-algorithms* attribute). (See the “[Creating and Editing Prefixes](#)” section on page 28-25.) If you specify use of the interface-identifier, the server might still use randomly generated addresses if the address is not available to the client, or the client requests multiple addresses on a prefix.

The server generates addresses based on the prefix-configured range (or the prefix address if there is no range). If the range prefix length is shorter than 64, the server supplies only 64 bits and places them in the address interface-identifier field. If the prefix length is longer than 64, the server supplies only the remaining bits of the address. Thus, a /96 range uses 96 bits from the specified range followed by 32 bits of either the client interface-identifier or a randomly generated value. If the resulting address is not available (such as if it is already leased to another client, or to the same client, but on a different binding), the server tries to generate another address. It repeats this process up to at most 500 times.

When DHCP failover is configured, the server generated addresses are always odd addresses on the main and even addresses on the backup.



Note

The DHCP server tests only the randomly generated interface identifier for values from `::0` to `::ff`, not the resulting address. Thus, a randomly generated address may end up using an `xxxx:xxxx:xxxx:xxxx::0` through `xxxx:xxxx:xxxx:xxxx::ff` address if the length of the prefix is longer than /64 and the prefix bits that extend beyond the /64 boundary are all zero.



Tip

You can also choose from additional address generation algorithms for a prefix and prefix template; see the “[Creating and Editing Prefix Templates](#)” section on page 28-20.

Generating Delegated Prefixes

The DHCPv6 server uses the best first-fit algorithm when generating delegated prefixes. The server uses the first longest available prefix of the length configured or requested.

For DHCP failover, each server only considers the delegated prefix leases in the available state. When the server is in the PARTNER-DOWN state, the server can also use leases in the other-available or pending-available states after certain time restrictions have passed.

Prefix Stability

Prefix Stability is introduced in Cisco Prime Network Registrar 8.1 to let you control prefix delegation independent of the network topology. A new link attribute *type* specifies the type of link.

There are three different link types:

- Topological—This is the link type supported in 8.0 and earlier. A client on a topological link is allocated leases based on the network segment it is connected to.
- Location independent—This link type is introduced to support the CableLabs DOCSIS 3.0 concept of CMTS prefix stability. It supports service provider load balancing and reconfiguration events within a group of CMTS (such as in a central office). A subscriber that is moved from one CMTS to another on a location-independent link can retain a delegated prefix. This link type allows movement within a single DHCP server.
- Universal—This link type is introduced to let subscribers retain a delegated prefix anywhere in the network. Use of this link type requires administrative assignment of the delegated prefixes and use of client or lease reservations. It can be deployed across multiple DHCP servers.

**Note**

Use of prefix stability has routing implications and requires appropriate support from relay agents (that is, CMTS) in order to advertise the routes. For CMTS prefix stability, these are localized to the CMTS group. The implications are greater for universal prefix stability as routes need to be advertised throughout the service providers network.

CMTS Prefix Stability

Location independent links implement the CableLabs DOCSIS 3.0 requirements for CMTS prefix stability. CMTS prefix stability is possible as long as all prefixes are serviced by a single DHCP server.

If you want to introduce CMTS prefix stability in a particular area, you need to:

- Modify existing links to specify the same link group name across all of the links within the group. Each CMTS (or CMTS bundle) will have a separate link, but all of these links within the area for which CMTS Prefix Stability is desired need to be made part of the same link group.
- Create a new link, flagged as location-independent and made part of this link group. Create or move one or more prefix delegation prefixes under this location-independent link - these are the prefixes from which the stable prefixes will be allocated.
- Remove any prefix delegation prefixes from the existing links that are no longer needed. Note that stateful prefixes (dhcp-type of dhcp) should not be removed.

**Note**

You can have only one location independent link in a group.

When a client request is received, the server locates the link by checking for the longest matching prefix and using the link of the prefix. However, if this topological link is part of a link group and that group has a location-independent link, the prefixes under the location-independent link will be checked first for possible leases requested by the client. Only if no leases are available from this location-independent link will the topological link be used. This is used for each binding requested by the client.

Any leasing mechanism (lease or client reservations, first best-fit, or extension generated/supplied) may be used with CMTS Prefix Stability as the leases are only known within the single server that services the CMTS group.

Universal Prefix Stability

Universal Prefix Stability lets you retain a delegated prefix regardless of where you connect. To use this feature, you must configure reservations for the delegated prefixes. Either client and lease reservations can be used.

Client reservations let you specify the delegated prefixes in a central LDAP repository that the DHCP servers access dynamically (see [“Using Client Reservations”](#) section on page 24-12). Lease reservations are managed centrally on the CCM regional server, and are pushed to each local DHCP with the universal link. Because the complete list of reservations is replicated on each server when using lease reservations, you should consider client reservations for larger deployments.



Note

You can have only one universal link in a particular VPN address space.

If a link is configured with the universal link type, the prefixes in that link are considered first when attempting to allocate a lease for a client. If no lease is available, the prefixes in the location-independent link type from the link group (if any) is used. Finally, the prefixes in the topological link are used.



Note

You can enable both CMTS Prefix Stability and Universal Prefix Stability at the same time, though only one will apply to a subscriber.

Prefix Allocation Groups

Cisco Prime Network Registrar 8.1 introduces prefix allocation groups to let you define multiple prefixes that do not result in multiple lease assignments to clients, and control the order in which the prefixes are used. The `allocation-group` and `allocation-group-priority` attributes are introduced to specify this behavior.

All prefixes on a link with the same allocation group name belong to that allocation group. A prefix with no allocation group name is in its own allocation group. At most one lease per binding is allocated across all the prefixes in the same allocation group.

The `allocation-group-priority` setting controls which prefixes are used. Lower numeric values have higher priority, except for 0 (the default), which has the lowest possible priority. Prefixes with the same priority are ordered by the active lease count, where the prefix with the lowest count will have the highest priority.



Note

The `allocation-group` name is only specific to the link. Different links can reuse the same allocation group names.

Starting from Cisco Prime Network Registrar 8.2, to control the number of leases a client can obtain from an allocation group prefix you can set the *max-leases-per-binding* attribute for the DHCP policy. For example, if you *set max-leases-per-binding* as 1, the client can obtain only one lease from an allocation group prefix. In addition, if more than one lease is already allocated from the same allocation group prefix then the additional leases are revoked (usually the oldest lease is revoked).

DHCPv6 Clients and Leases

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

- The server identifies DHCPv6 clients 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 and are not grouped using the *allocation-group* attribute, the server assigns the client a lease from each prefix that it is allowed to use, not just from one scope, as in DHCPv4. If multiple prefixes on a single link are grouped using the *allocation-group* attribute, then the server assigns the client only one lease per allocation group from the prefix with highest priority within the prefix allocation group (see “[Prefix Allocation Groups](#)” section on page 28-6).
- The server first creates a DHCPv6 client when it associates the first lease with it, and deletes the client 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. The server does not create all leases that it can potentially use at configuration time, because there potentially could 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 use of the client, 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.

Related Topics

[DHCPv6 Bindings](#)

[Lease Affinity](#)

[IPv6 Lease States, page 28-8](#)

[Lease Life Cycle, page 28-9](#)

[DHCPv6 Lease Reservations, page 28-10](#)

[DHCPv6 Client Reservations, page 28-12](#)
[Searching for Leases, page 28-12](#)
[Querying Leases for DHCPv6, page 28-12](#)

DHCPv6 Bindings

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

- Nontemporary (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. The server creates bindings when it first adds the lease, and removes the binding when it has no more leases. The server creates clients when adding the first binding, and removes them when it has no more bindings.

Lease Affinity

For DHCPv4, when a lease expires or the server releases it, the server remembers the client for an address as long as it is not assigned to another client. For DHCPv6, because of the large IPv6 address space and depending on the address generation technique, eons could pass before an address needs reassignment to another client. Therefore, Cisco Prime Network Registrar provides an *affinity-period* attribute so that the client can get the same address even if not requesting a renewal before expiration.

The affinity period is desirable in some environments, but not in others where the affinity time would be zero or very small. 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 period, the server grants it the same lease (or, if renewals are inhibited, the client explicitly does not get that lease).

IPv6 Lease States

A lease can be in one of the states described in [Table 28-1](#).

Table 28-1 IPv6 Lease States

State	Description
Available	IP address available to be leased.
Offered	Offered to the client.
Leased	Held by a client.
Expired	Available when the lease grace period expires.
Unavailable	Not leasable. It was made unavailable because of some conflict.
Released	The client has released the lease, but the server is configured to apply a grace period to the lease. The lease will not be made available until the grace period expires.
Other available	Failover-related. Available for allocation by the failover partner but not available for allocation by this server.

Table 28-1 IPv6 Lease States (continued)

State	Description
Pending available	Failover-related. A lease in the pending-available state is available as soon as the server synchronizes its state with the failover partner. Used for only prefix delegation leases.
Pending delete	Failover-related. A lease in the pending-delete state is disassociated from its client as soon as the server synchronizes its state with the failover partner.

Lease Life Cycle

Leases have a life cycle controlled by states. A lease only exists while it is associated with a client and the server deletes it 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 the server:
 - a. Creates a reservation for a lease, which puts the lease in the AVAILABLE state and marks it as RESERVED. No timer is associated with this state and the server does not delete the lease as long as it is RESERVED.
 - b. Sends an ADVERTISE message to a client, which puts the lease in OFFERED state. The lease transitions to DELETED state after the offer timeout.
 - c. Sends a REPLY message to a client (for a REQUEST, RENEW, or REBIND), which puts the lease in LEASED state. The lease transitions to EXPIRED state after the valid lifetime for the lease elapses.
2. An OFFERED lease transitions to:
 - a. LEASED state when the server receives a REQUEST message, 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 the server receives a REQUEST, RENEW, or REBIND message. The lease 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 the server receives a RELEASE message. The lease transitions to AVAILABLE state after the release-grace-period elapses.
 - c. Transitions to UNAVAILABLE state when the server receives a DECLINE message. The server deletes the lease after the unavailable timeout period elapses.
4. An EXPIRED lease transitions to either AVAILABLE state after the grace-period. The server deletes the lease after the affinity-period elapses.
5. An AVAILABLE lease:
 - d. Transitions to DELETE state and the server deletes it from memory and the lease database after the affinity-period elapses.
 - e. Cannot be deleted if it is RESERVED, and it remains AVAILABLE.
6. The server can reoffer a LEASED, EXPIRED, RELEASED, or AVAILABLE lease to a client, but it remains in its current state, although the server extends the timeout to at least the *offer-timeout*.

The DHCP failover complicates some of the state transitions as these transitions can generally not occur until the partner acknowledges them. The additional life cycle and state transitions (failover related) are as follows:

- Transitioning into the AVAILABLE (or OTHER AVAILABLE) state requires that the partner to acknowledge the transition and hence the PENDING AVAILABLE state is used until the acknowledgement is received from the partner.
- Disassociating a lease from a client also requires an acknowledgement from the partner and hence the PENDING DELETE state is used until the partner has acknowledged the state change.

DHCPv6 Lease Reservations

Reservations apply to nontemporary addresses and delegated prefixes only. They are associated with a prefix in the configuration, and must always be for an address (or prefix) under a configured prefix object.

The reservation can be outside the object range of the prefix, provided it is not in object range of another prefix. However, this has implications when you add a new prefix object. If a reservation that is contained in the new range of the prefix exists, the prefix will not be added. This results in an EX_CONFLICT status. For details, see the [“Creating Lease Reservations” section on page 24-15](#).



Note

The operations for DHCPv4 reservations are similar to DHCPv6 reservations, except that the addresses are v6 addresses, not v4 addresses. Also, the main identity for a DHCPv6 client is a client DUID, and not the mac-address. DHCPv6 reservations include addresses and delegated prefixes.

Any change you make in the v6 reservation list modifies the parent prefix to indicate that a server reload is required. On the regional server, if the DHCP edit mode is synchronous and the parent prefix has been assigned to a local cluster, changes are automatically forwarded to the local cluster. A server reload is required, before these changes take effect.



Caution

If multiple DHCP servers distribute IP addresses on the same prefix, the reservations must be identical. If not, a client for whom a reservation exists can receive offers of different IP addresses from different servers.

A lease reservation pairs an IP address with a lookup key. A lookup key can be a string value or binary blob.



Note

If a new prefix delegation reservation is added that has a shorter or longer prefix that conflicts (is contained by or contained in) an existing lease when the server is reloaded, the reservation will prevent the existing leases from being loaded.

Local Advanced Web UI

To view the reservations for DHCPv6 prefixes, do the following:

- Step 1** To view the DHCPv6 lease reservations, from the **Design** menu, choose **Prefixes** under the **DHCPv6** submenu to open the List/Add DHCPv6 Prefixes page.
- Step 2** Select the prefix on the Prefixes pane and click the **Reservations** tab.

To configure the reservations directly for DHCPv6 prefixes, do the following:

In the advanced mode, if a valid parent prefix is not specified, the CCM server automatically sets the appropriate parent prefix.

-
- Step 1** From the **Design** menu, choose **Reservations** under the **DHCPv6** submenu to open the List/Add DHCPv6 Reservations page.
 - Step 2** To create a reservation, click the **Add Reservations** icon in the Reservations pane, enter the IP address you want to reserve for lease, and enter a lookup key in the Lookup Key field.
 - Step 3** Click the **String** radio button, if you entered string value or click the **Binary** radio button, if you entered binary value in the Lookup Key field.
 - Step 4** Click **Add v6 Reservation**.
 - Step 5** On the Reservations pane, choose a filter type from the Filter Type drop-down list. Enter a value in the Filter Value field. Click **Set Filter**. To set Filter Type as None, click **Clear Filter**. The lease IP address, Lookup Key and Prefix details are displayed in the List/Add DHCPv6 Reservations page.
-

CLI Commands

The reservation6 command lets you access the global list of DHCPv6 reservations of Cisco Prime Network Registrar.

A matching prefix must exist for each reservation in the global list, otherwise the edit is rejected as invalid.

Create a new address by using, **reservation6** [vpn-name/]address **create** lookup-key [**blob** | **string**]

For example:

```
nrcmd> reservation6 white/2001:db8::1 create 00:03:00:01:01:02:03:04:05:06
```

Delete an address by using, **reservation6** [vpn-name/]address **delete**

For example:

```
nrcmd> reservation6 white/2001:DB8::1 delete
```

Get an address by using, **reservation6** [vpn-name/]address **get** value

For example:

```
nrcmd> reservation6 white/2001:DB8::1 get value
```

Set an attribute by using, **reservation6** [vpn-name/]address **set** prefix=<name-of-prefix>

For example:

```
nrcmd> reservation6 white/2001:DB8::1 set prefix=cm_prefix
```

Unset an attribute by using, **reservation6** [vpn-name/]address **unset** value

For example:

```
nrcmd> reservation6 white/2001:DB8::1 unset value
```

Show an address by using, `reservation6 [vpn-name/]address show`

For example:

```
nrcmd>reservation6 white/2001:DB8::1 show
```

DHCPv6 Client Reservations

Cisco Prime Network Registrar supports both Lease and Client Reservations for DHCPv6. For details on Client Reservations, see [“Using Client Reservations” section on page 24-12](#).

Searching for Leases

For details on searching for leases in the configured DHCPv6 network, see the [“Searching Server-Wide for Leases” section on page 24-9](#).

Querying Leases for DHCPv6

For details on the DHCPLEASEQUERY implementation for DHCPv6, see the [“Leasequery for DHCPv6” section on page 24-34](#).

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

For attributes, the default value for the most local policy applies. 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 [“Policy Hierarchy” section on page 23-3](#).)

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

- For the *reconfigure* attribute that controls whether the server requires, allows, or disallows client reconfiguration, the server checks the embedded and named policies of all prefixes on the link that the client is allowed to use (based on selection tags). If any of the prefix policies have the *reconfigure* attribute set to **disallow** or **require**, the server uses that setting. Otherwise, if at least one policy has

it set to **allow**, Reconfigure is allowed. Otherwise, the server checks the remaining policies in the hierarchy. (See the “[Reconfigure Support](#)” section on page 28-35 for details.)

- If the client requests Rapid Commit (see the “[Editing DHCPv6 Server Attributes](#)” section on page 28-32), the server checks the embedded and named policies of all 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 server processes the client request as if Rapid Commit were not part of the request. If at least one policy has *allow-rapid-commit* enabled, the client can use Rapid Commit. If no prefix policy has the attribute configured, processing continues at step 7.
- For the options-related attributes (see the “[Setting DHCPv6 Options](#)” section on page 28-35), the server also does special handling at steps 5 and 6. The server checks the embedded and then named policy of each 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 server checks the prefixes in case-insensitive alphabetical order.
- The server ignores any policies related to the location-independent and/or universal link and the prefixes under those. Only topological links (and prefixes under those links) are considered.

**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 DHCPv4 options; they are unique and separate. There are currently about 46 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). Configuring options and the behavior of configured options in policies are similar to those for DHCPv4. See the “[Setting DHCPv6 Options](#)” section on page 28-35 for details about client processing as it relates to the policy hierarchy.

DHCPv6 Configuration

The following sections describe how to configure DHCPv6 in Cisco Prime Network Registrar:

- [Viewing IPv6 Address Space](#)
- [Configuring Links](#)
- [Configuring Prefixes, page 28-20](#)
- [Viewing DHCPv6 Networks, page 28-32](#)
- [Editing DHCPv6 Server Attributes, page 28-32](#)
- [Configuring DHCPv6 Policies, page 28-33](#)
- [Configuring DHCPv6 Client-Classes, page 28-34](#)
- [Configuring DHCPv6 Clients, page 28-34](#)
- [Setting DHCPv6 Options, page 28-35](#)
- [Reconfigure Support, page 28-35](#)

Viewing IPv6 Address Space

From the **Design** menu, choose **Address Tree** under the **DHCPv6** submenu, in the local advanced or regional web UI, to view the DHCP v6 Address Tree page. This page is like the DHCP Address Tree page for IPv4 (see the “[Viewing Address Space](#)” section on page 9-3). On the View Unified v6 Address Space page you can:

- Set a VPN for the address space from the *username* drop-down list on the top right of the window.
- Add a prefix by clicking the **Add Address Tree** icon in the Address Tree pane, enter the prefix name, address and choosing a DHCP type and possible template. Click **Add IPv6 Prefix** (see the “[Creating and Editing Prefixes](#)” section on page 28-25).
- Edit a prefix by selecting its name in the Address Tree pane. This opens the Edit IPv6 Prefix page (see the “[Creating and Editing Prefixes](#)” section on page 28-25).
- View the current usage of the prefix space (see the “[Viewing Address Utilization for Prefixes](#)” section on page 28-30).

Configuring Links

You can configure DHCPv6 links directly, or you can create link templates for them first. See the following subsections:

- [Creating and Editing Link Templates](#)
- [Creating and Editing Links](#), page 28-18

Creating and Editing Link Templates

You can create links from predefined templates. The attributes you can set for a link template are as follows (for the expression syntax, see the “[Using Expressions in Link Templates](#)” section on page 28-16):

- *name*—User-assigned name for the link template.
- *description*—Description of the link template itself.
- *policy*—Shared policy used when replying to clients, as applied to the link.
- *owner*—Owner of the link.
- *region*—Region for this link.
- *link-name-expr*—Expression to define the name of the link once the template is applied.
- *link-description-expr*—Expression to define the description on the link once applied.
- *prefix-expr*—Expression to create the list of associated prefixes once the template is applied. For example, you can specify creating prefixes based on defining *prefix-expr* as **@link-prefix-expr.txt** to point to the file that contains this expression (and assuming that the cm-prefix, cpe-address-prefix, and cpe-pd-prefix templates exist):


```
(list
  (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
  (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
  (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1))
)
```
- *options-expr*—Expression to define the list of embedded policy options to create with the link.

- *free-addr-config*—Trap that captures unexpected free address events on this link
- *type*—Type of the link (topological, location-independent, universal).
- *group-name*—Link group to which the link belongs.

Local Advanced and Regional Web UI

-
- Step 1** From the **Design** menu, choose **Link Templates** under the **DHCPv6** submenu. The List/Add DHCP v6 Link Templates page appears. The page displays the existing templates.
- Step 2** Click the **Add Link Templates** icon in the Link Templates pane to open the Add Link Template dialog box.
- Step 3** Enter a link template name and click **Add Link Template**.
- Step 4** Enter an optional description, and optionally choose a preconfigured policy from the drop-down list.
- Step 5** Add expressions for the *link-name-expr*, *link-description-expr*, *prefix-expr*, or *options-expr* field attributes (see the [“Using Expressions in Link Templates”](#) section on page 28-16).
- Step 6** If the link template is for Prefix Stability, select the link type (type) and specify a link group name (group-name). You can find these attributes in the Prefix Stability block in the Add DHCP v6 Link Template page (see the [“Prefix Stability”](#) section on page 28-5 for details on link types and link groups).
- Step 7** Click **Save**.
- Step 8** In the regional web UI, you can pull replica link templates or push templates to local clusters:
- Click **Pull Replica Link Template** to open the Select DHCPv6 Link Template Data to Pull page. Choose a pull mode for the cluster (ensure, replace, or exact), then click **Pull All Link Templates**. On the Report Pull DHCPv6 Link Template page, click **OK**.
 - Click **Push Link Template** for a specific template (or **Push All Link Templates**) to open the Push DHCPv6 Link Template Data to Local Cluster page. Choose a data synchronization mode (ensure, replace, or exact), move the desired cluster or clusters to the Selected table, then click **Push Data to Clusters**.
-

CLI Commands

To create the link template, use **link-template name create**. For example:

```
nrcmd> link-template example-link-template create [attribute=value]
```

You can set and enable the aforementioned expression setting attributes in the usual way, and you can show and list link templates. For example, to set a prefix expression for the link template, use the following file definition and pointer to the file (and assuming that the cm-prefix, cpe-address-prefix, and cpe-pd-prefix templates exist):

```
> type link-prefix-expr.txt
(list (create-prefix "cm-prefix" (create-prefix-range 32 0x1))
 (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2))
 (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1)) )
```

```
nrcmd> link-template example-link-template set prefix-expr=@link-prefix-expr.txt
```

In addition:

- To clone a link template, use **link-template name create clone=name**.

- To apply a template to one or more links, use **link-template name apply-to {all | link[,link,...]}**. You can create prefixes by using **link-template name apply-to link [prefix]**, but only with one link specified.
- The link-template includes an embedded-policy object. The link-template-policy CLI command and the Web UI supports the embedded policy on the link-template page.

Using Expressions in Link Templates

You can specify expressions in a link template to dynamically create prefix names, IP address ranges, and embedded options when creating a link. Expressions can include context variables and operations.



Note

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see [Chapter 31, “Using Extension Points”](#)) are used to modify request or response packets.

When a template is applied to a link, if the link-template has an embedded policy, it is copied to the link. This embedded policy may or may not have options. As the entire link-template’s embedded policy is used (if it exists), it will wipe out any existing options in the link. If the link-template has no embedded policy, the link’s embedded policy is retained. Next the link-template’s option expression, if any, is evaluated and the options are added to the embedded policy options in the link (if no embedded policy exists, one is created).

[Table 28-2](#) lists the link template predefined variables and [Table 28-3](#) lists the link template operators. Note that these variables and operators are not case-sensitive. [Table 28-5](#) lists the prefix template operators. The link template operators table and prefix template operations table both have same operators, except that only a link template can use **Create Prefix Operator** and prefix template can not use the operator.

Table 28-2 Link Template Expression Predefined Variables

Predefined Variable	Description
mask-length	Number of prefix mask bits (with a <i>template-root-prefix</i> defined).
prefix	Network number and length (with a <i>template-root-prefix</i> defined).
prefix-addr	Address portion of the prefix (with a <i>template-root-prefix</i> defined).
prefix-length	Number of prefix address bits (with a <i>template-root-prefix</i> defined).
template.attribute	Attribute of the link template.
vpn	VPN of the link.



Table 28-3 Link Template Expression Operators

Expression Operator	Description
Arithmetic Operators (unsigned integer arguments only)	
(+ arg1 arg2)	Adds the two argument values, such as (+ 2 3).
(- arg1 arg2)	Subtracts the second argument value from the first one.
(* arg1 arg2)	Multiplies the values of two arguments.
(/ arg1 arg2)	Divides the value of the first argument by that of the second one (which cannot be zero).

Table 28-3 Link Template Expression Operators (continued)

Expression Operator	Description
<code>(% arg1 arg2)</code>	Modulo arithmetic operator to determine the remainder of the result of the first argument divided by the second one.
Concatenation Operator	
<code>(concat arg1 ... argn)</code>	Concatenates the arguments into a string.
List Operator	
<code>(list oper1 ... opern)</code>	Creates an options list or list of prefixes. Required if you need more than one option for a link or prefix, or more than one prefix for a link. All arguments must be create-v6-option operation. Nesting is not supported. For example: <pre>(list (create-prefix "cm-prefix" (create-prefix-range 32 0x1)) (create-prefix "cpe-address-prefix" (create-prefix-range 32 0x2)) (create-prefix "cpe-pd-prefix" (create-prefix-range 16 0x1)))</pre>
Create Prefix Operator	
<code>(create-prefix template prefix)</code>	Creates a prefix based on a predefined prefix template name and the prefix, including the link VPN (assuming that a <i>template-root-prefix</i> is defined). The <i>prefix</i> argument can be the prefix name, but also the create-prefix-addr or create-prefix-range operator value. You can use the list function to combine multiple operations. For example: <pre>(create-prefix "cm-prefix" (create-prefix-range 32 0x1))</pre>
Create IP Operator	
<code>(create-prefix-addr prefix interface-id)</code>	Creates an IPv6 address string (assuming that a <i>template-root-prefix</i> is defined) based on the prefix name and interface ID (an IPv6 address that you can specify as a string), which is the lower 64-bit address in the prefix (which need not be contained in the parent prefix). Used in the <i>prefix-expr</i> and <i>options-expr</i> attributes.
Create Range Operator	
<code>(create-prefix-range size n)</code>	Creates an address range (child) for the prefix, used in the <i>prefix-expr</i> attribute. The prefix value that the function is based on is either the <i>template-root-prefix</i> if applying a link template to a link, or the prefix address, if applying a prefix template to a prefix. Range value—An increase in the prefix length. Size—The number of bits by which you can increase the prefix length. Must be a value from 1 through 32. Must be less than the parent prefix length. n—The nth occurrence of the child prefix. Value can be 0, but is limited to less than two to the power of the size. Must be less than or equal to the size. The size and n must be greater than zero. The <i>n</i> must be less than or equal to the <i>size</i> , and the <i>size</i> must be less than the parent prefix length. For example: <pre>(create-prefix-range 32 0x1)</pre>

Table 28-3 Link Template Expression Operators (continued)

Expression Operator	Description
Create Option Operator	
(create-option <i>opt val</i>)	<p>Creates a DHCPv6 option, used in the <i>options-expr</i> attribute. The <i>opt</i> can be the literal string or integer identifying the option. The <i>val</i> is the string representation of the option value, as defined by the option TLV value.</p> <p>You can use custom defined and unknown options. For undefined options, the option number must be specified and the data is used as is (as blob data). If the data is a string, the string is used as is and if the data is a number or address, it is used as is.</p> <p>For example:</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::<2")) (create-option "domain-list" "sales.example.com,example.com"))</pre>
	<p>Note (create-v6-option <i>opt val</i>) is a synonym for (create-option) and can be used instead; but we recommend that you use (create-option).</p>
Create Vendor Option Operation	
(create-vendor-option <i>set-name opt val</i>)	<p>Creates a DHCPv6 vendor option, used in the <i>options-expr</i> attribute. The <i>set-name</i> specifies the option definition set for the vendor option. The <i>opt</i> can be the literal string or integer identifying the vendor option in the set. The <i>val</i> is representation of the option value.</p> <p>For example:</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::<2")) (create-vendor-option "dhcp6-cablelabs-config" 17 "(enterprise-id 4491((tftp-servers 32 3800:0:0:180::6) (config-file-name 33 modem_ipv6.bin)(syslog-servers 34 3800:0:0:180::8) (rfc868-servers 37 3800:0:0:180::6)(time-offset 38 -5h) (cablelabs-client-configuration 2170 (primary-dhcp-server 1 10.38.1.5) (secondary-dhcp-server 2 10.38.1.6))))))</pre>
	<p>Note (create-v6-vendor-option <i>opt val</i>) is a synonym for (create-vendor-option) and can be used instead; but we recommend that you use (create-vendor-option).</p>

Creating and Editing Links

You can create links directly. The attributes you can set for the link are:

- *name*—User-assigned name for the link.
- *vpn-id*—VPN that contains the link.
- *description*—Descriptive text for the link.
- *policy*—Shared policy used when replying to clients.
- *owner*—Owner of the link.

- *region*—Region for this link.
- *free-address-config*—Identifies which trap captures unexpected free address events on this prefix. If not configured, the server looks at its *v6-default-free-address-config* attribute.
- *type*—Type of link (topological, location-independent, universal).
- *group-name*—Link group to which the link belongs.

Local Advanced and Regional Web UI

-
- Step 1** From the **Design** menu, choose **Links** under the **DHCPv6** submenu. The List/Add DHCP v6 Links page displays the existing links.
- Step 2** To add a link, click the **Add Link** icon in the Links pane.
- Step 3** Enter the desired name for the link.
- Step 4** If the link is for Prefix Stability, select the link type (*type*) and specify a link group name (*group-name*). The Link Type is *topological*, by default. You can also find these attributes in the Prefix Stability area in the Edit DHCP v6 Link Template page (see the “[Prefix Stability](#)” section on page 28-5 for details on link types and link groups).



Note You can have only one location independent link in a link group and one universal link in a VPN address space. Also, you cannot assign a link of type universal to a link group.

- Step 5** Click **Add Link**.
- Step 6** In the Edit Link page of the new link, choose the predefined prefixes for the link by moving them from the Available field to the Selected field.
- Step 7** To add new prefixes for the link, enter each prefix name and its address at the bottom of the page, indicate a range, choose the DHCP type and template (if needed), then click **Apply Prefix** for each one.
- Step 8** Click **Save**.
- Step 9** In the regional web UI, you can push links to local clusters and reclaim links on the Edit DHCP v6 Link page and pull replica IPv6 address space on the List/Add DHCP v6 Links page:
- To push the link, click **Push** (at the top of the page) to open the Push DHCP v6 Link page. Choose the cluster or link template to which you want to push the link, then click **Push Link**. When the link is pushed, all prefixes on the link, and all reservations on the prefixes are also be pushed.
 - To reclaim the prefix, click **Reclaim** (at the top of the page) to open the Reclaim DHCP v6 Link page. Choose the cluster or link template to which you want to reclaim the link, then click **Reclaim Link**. When the link is reclaimed, the reservations, prefixes, and link is deleted from the local cluster, provided there are no active leases. If active leases are found, prefixes are deactivated instead. The force option lets you remove the link and its prefixes when there are active leases.



Note Only universal links can be pushed to more than one cluster.

- To pull replica IPv6 address space, click the **Pull Replica** icon (at the top of the links pane on the left) to open Select Pull Replica IPv6 Address Space. Choose the data synchronization mode (update, complete, or exact) and click **Report**.

The local changes will not take effect until the next server reload.

CLI Commands

Use **link name create**. (The **link** command is a synonym for the **dhcp-link** command from previous releases.) For example:

```
nrcmd> link example-link create [attribute=value]
```

To apply a link template during link creation, use **link name create template=name [template-root-prefix=address]**, with the *template-root-prefix* specified if the template could create more than one prefix. To apply a template to an existing link definition, use **link name applyTemplate template-name [template-root-prefix]**.

You can set and enable the aforementioned attributes in the usual way, and you can show and list links. To list prefixes or prefix names associated with a link, use **link name listPrefixes** or **link name listPrefixNames**.

Configuring Prefixes

You can configure DHCPv6 prefixes directly, or you can create prefix templates for them first. See the following subsections:

- [Creating and Editing Prefix Templates](#)
- [Creating and Editing Prefixes, page 28-25](#)
- [Viewing Address Utilization for Prefixes, page 28-30](#)

Creating and Editing Prefix Templates

You can create prefixes from predefined templates. The attributes you can set for a prefix template are the following (for the expression syntax, see the “[Using Expressions in Prefix Templates](#)” section on [page 28-23](#)):

- *name*—User-assigned name for the prefix template.
- *description*—Descriptive text for the prefix template.
- *dhcp-type*—Defines how DHCP manages address assignment for a prefix:
 - **dhcp** (preset value)—Uses the prefix for stateful address assignment.
 - **stateless**—Uses the prefix for stateless option configuration.
 - **prefix-delegation**—Uses the prefix for prefix delegation.
 - **infrastructure**—Uses the prefix to map a client address to a link, when the prefix does not have an address pool.
- *policy*—Shared policy to use when replying to clients.
- *owner*—Owner of this prefix, referenced by name.
- *region*—Region for this prefix, referenced by name.
- *prefix-name-expr*—Expression that evaluates to a string value to use for the name of the prefix created. For example, you can have the prefix name prepended by **CM-** if you define *prefix-name-expr* as (**concat "CM-" prefix**). In the CLI, you would include the expression in a file and point to that file:

```
> type prefix-name.txt
(concat "CM-" prefix)
```

```
nrcmd> prefix-template ex-template create prefix-name-expr=@prefix-name.txt
```

- **prefix-description-expr**—Expression that evaluates to a string value to apply to the description on the prefix created when using the template.
- **range-expr**—Expression that evaluates to an IPv6 prefix value to create an address range. In the CLI, you must use a file reference. For example:

```
> type subprefix-expr.txt
(create-prefix-range 1 0x1)
```

```
nrcmd> prefix-template ex-template set range-expr=@subprefix-expr.txt
```

- **options-expr**—Expression that evaluates to embedded policy options to create. (Use the **list** function to create multiple options.)
- **allocation-algorithms**—One or more algorithms the server uses to select a new address or prefix to lease to a client. The available algorithms are:
 - **client-request** (preset to off)—Controls whether the server uses a client-requested lease.
 - **reservation** (preset to on)—Controls whether the server uses an available reservation for the client.
 - **extension** (preset to on)—Controls whether the server calls extensions attached at the **generate-lease** extension point to generate an address or prefix for the client. When you use generate-lease extension point with DHCPv6 failover, the server uses the address or delegated prefix that the extension returns and does not perform a hash on this address or prefix as it does with the randomly generated addresses. If the extension is using some algorithmic method to generate the address or delegated prefix then the extension must be failover aware (extension will be able to determine if failover configuration is enabled and the role of the failover server). For details on extensions, see [Chapter 31, “Using Extension Points.”](#)
 - **interface-identifier** (preset to off)—Controls whether the server uses the interface-identifier from the client (link-local) address to generate an address; ignored for temporary addresses and prefix delegation.
 - **random** (preset to on)—Controls whether the server generates an address using an RFC 3041 algorithm; ignored for prefix delegation.
 - **best-fit** (preset to on)—Controls whether the server delegates the first, best-fit available prefix; ignored for addresses.

When the server needs an address to assign to a client, it processes the flags in the following order until it finds a usable address: client-request, reservation, extension, interface-identifier, and random. When the server needs to delegate a prefix to a client, it processes the flags in the following order until it finds a usable prefix: client-request, reservation, extension, and best-fit.

- **restrict-to-reservations**—Controls whether the prefix is restricted to client (or lease) reservations.
- **max-leases**—Maximum number of nonreserved leases allowed on the prefix. When a new lease needs to be created, the server does so only if the limit is not exceeded. When the limit is exceeded, the server cannot create or offer new leases to clients. If you also enable SNMP traps, the *max-leases* value also calculates the percentage of used and available addresses.



Note Be sure to set the *max-leases* value to the expected maximum so that the SNMP address traps can return meaningful results.

- **ignore-declines**—Controls whether the server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix. If enabled, the server ignores all declines for leases in this prefix. If disabled (the preset value) or unset, the server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE message if it is leased to the client.
- **deactivated**—Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients and treats all addresses in its ranges as if they were individually deactivated. The preset value is false (activated).
- **expiration-time**—Time and date at which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix. Enter a value in the format "[weekday] month day hh:mm[:ss] year"; for example, "**Dec 31 23:59 2006**". The reason for an expiration time is to support network renumbering events. The general idea is a new prefix is added and the old is taken away sometime at or after the *expiration-time*. Clients will be given leases on both prefixes. The server will automatically stop giving new clients leases once the configured valid lifetime before the expiration-time is reached. At this time, new clients will not get a lease on the prefix. Existing clients will continue to be able to use an existing lease, but will get shorter and shorter lifetimes (preferred and valid). The delta between the preferred and valid is always maintained. Thus if the preferred is 1day and the valid 2days, new clients will stop getting leases 2 days before the *expiration-time*, existing clients will continue to be able to renew leases with preferred lifetimes lesser than 1day and valid lifetimes greater than 2days. 1 day before the *expiration-time*, clients will get a 0 preferred lifetime.
- **free-addr-config**—Trap that captures unexpected free address events on the prefix.
- **reverse-zone-prefix-length**—Prefix length of the reverse zone for ip6.arpa updates. (See the “Determining Reverse Zones for DNS Updates” section on page 30-4 for details.)
- **max-pd-balancing-length**—Controls the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix. The default value is 64 and it should never be longer than the longest prefix length allowed for the prefix delegation.
- **selection-tags**—List of selection tags associated with the prefix.
- **allocation-group**—Allocation group to which the prefix belongs.
- **allocation-group-priority**—Priority of the prefix over other prefixes in the same allocation group. The default value is zero.

Local Advanced and Regional Web UI

-
- Step 1** From the **Design** menu, choose **Prefix Templates** under the **DHCPv6** submenu. The List/Add DHCP v6 Prefix Templates page shows the existing templates.
 - Step 2** Click the **Add Prefix Templates** icon in the **Prefix Templates** pane to open the Add Prefix Template dialog box.
 - Step 3** Enter the prefix template name and click **Add Prefix Template**.
 - Step 4** To edit a prefix template, select its name on the Prefixes pane. Set the attributes and add expressions for the templates that require expressions (see the “Using Expressions in Prefix Templates” section on page 28-23).
 - Step 5** On the Edit DHCPv6 Prefix Template page, edit the template attributes, such as adding a selection tag, assigning a group and setting priorities, then click **Save**.
 - Step 6** In the regional web UI, you can pull replica prefix templates or push templates to local clusters:

- Click **Pull Replica Prefix Template** to open the Select DHCPv6 Prefix Template Data to Pull page. Choose a pull mode for the cluster (ensure, replace, or exact), then click **Pull All Prefix Templates**. On the Report Pull DHCPv6 Prefix Template page, click **OK**.
- Click **Push Prefix Template** for a specific template (or **Push All Prefix Templates**) to open the Push DHCPv6 Prefix Template Data to Local Cluster page. Choose a data synchronization mode (ensure, replace, or exact), move the desired cluster or clusters to the Selected table, then click **Push Data to Clusters**.

CLI Commands

To create the prefix template, use **prefix-template name create**. For example:

```
nrcmd> prefix-template example-prefix-template create [attribute=value]
```

You can set and enable the aforementioned attributes in the usual way, and you can show and list prefix templates. In addition:

- To clone a prefix template, use **prefix-template name create clone=name**.
- To apply a template to one or more prefixes, use **prefix-template name apply-to {all | prefix[,prefix,...]}**.
- The prefix-template includes an embedded-policy object. The prefix-template-policy CLI command and the Web UI supports the embedded policy on the prefix-template page.

Using Expressions in Prefix Templates

You can specify expressions in a prefix template to dynamically create prefix names, IP address ranges, and embedded options when creating a prefix. Expressions can include context variables and operations.



Note

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see [Chapter 31, “Using Extension Points”](#)) are used to modify request or response packets.

When a template is applied to a prefix, if the prefix-template has an embedded policy, it is copied to the prefix. This embedded policy may or may not have options. As the entire prefix-template’s embedded policy is used (if it exists), it will wipe out any existing options in the prefix. If the prefix-template has no embedded policy, the prefix’s embedded policy is retained. Next the prefix-template’s option expression, if any, is evaluated and the options are added to the embedded policy options in the prefix (if no embedded policy exists, one is created).

[Table 28-4](#) lists the prefix template predefined variables and [Table 28-5 on page 28-24](#) lists the operators. Note that these variables and operators are not case-sensitive.

Table 28-4 Prefix Template Expression Predefined Variables

Predefined Variable	Description
prefix	Network number and length, based on the template root prefix if applying a link template to a link, or the prefix address if applying a prefix template to a prefix.
vpn	VPN of the prefix.
prefix-addr	Address portion of the prefix.



Table 28-4 Prefix Template Expression Predefined Variables (continued)

Predefined Variable	Description
prefix-length	Number of prefix address bits.
mask-length	Number of prefix mask bits.
template.attribute	Attribute of the prefix template.

Table 28-5 Prefix Template Expression Operators

Expression Operator	Description
Arithmetic Operators (unsigned integer arguments only)	
(+ arg1 arg2)	Adds the two argument values, such as (+ 2 3).
(- arg1 arg2)	Subtracts the second argument value from the first one, such as with <i>ping-timeout</i> defined as 100, (- template.ping-timeout 10) yields 90.
(* arg1 arg2)	Multiplies the values of two arguments.
(/ arg1 arg2)	Divides the value of the first argument by that of the second one (which cannot be zero).
(% arg1 arg2)	Modulo arithmetic operator to determine the remainder of the result of the first argument divided by the second one.
Concatenation Operator	
(concat arg1 ... argn)	Concatenates the arguments into a string.
List Operator	
(list oper1 ... opern)	Creates an options list or list of prefixes. Required if needing more than one option for a prefix. All arguments must be create-v6-option or create-prefix-range operations. Nesting is not supported.
Create IP Operator	
(create-prefix-addr prefix-name interface-id)	Creates an IPv6 address string based on the prefix name and interface ID (an IPv6 address that you can specify as a string), which is the lower 64-bit address in the prefix (which need not be contained in the parent prefix). Used in the <i>range-expr</i> and <i>options-expr</i> attributes.
Create Range Operator	
(create-prefix-range size n)	Creates an address range (child) for the prefix, used in the <i>range-expr</i> attribute. The prefix value that the function is based on is either the <i>template-root-prefix</i> if applying a link template to a link, or the prefix address if applying a prefix template to a prefix. Range value—An increase in the prefix length. Size—The number of bits by which you can increase the prefix length. Must be a value from 1 through 32. Must be less than the parent prefix length. n—The nth occurrence of the child prefix. Value can be 0, but is limited to less than two to the power of the size. Must be less than or equal to the size. The size and n must be greater than zero. The n must be less than or equal to the size, and the size must be less than the parent prefix length. For example: (create-prefix-range 32 0x1)

Table 28-5 Prefix Template Expression Operators (continued)

Expression Operator	Description
Create Option Operation	
(create-option <i>opt val</i>)	<p>Creates a DHCPv6 option, used in the <i>options-expr</i> attribute. The <i>opt</i> can be the literal string or integer identifying the option. The <i>val</i> is the string representation of the option value, as defined by the option TLV value.</p> <p>You can use custom defined and unknown options. For undefined options, the option number must be specified and the data is used as is (as blob data). If the data is a string, the string is used as is and if the data is a number or address, it is used as is.</p> <p>For example:</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::2")) (create-option "domain-list" "sales.example.com,example.com"))</pre>
	
Note	(create-v6-option <i>opt val</i>) is a synonym for (create-option) and can be used instead.
Create Vendor Option Operator	
(create-vendor-option <i>set-name opt val</i>)	<p>Creates a DHCPv6 vendor option, used in the <i>options-expr</i> attribute. The <i>set-name</i> specifies the option definition set for the vendor option. The <i>opt</i> can be the literal string or integer identifying the vendor option in the set. The <i>val</i> is representation of the option value.</p> <p>For example:</p> <pre>(list (create-option "dns-servers" (create-prefix-addr prefix "::2")) (create-vendor-option "dhcp6-cablelabs-config" 17 "(enterprise-id 4491((tftp-servers 32 3800:0:0:180::6) (config-file-name 33 modem_ipv6.bin) (syslog-servers 34 3800:0:0:180::8) (rfc868-servers 37 3800:0:0:180::6) (time-offset 38 -5h) (cablelabs-client-configuration 2170 (primary-dhcp-server 1 10.38.1.5) (secondary-dhcp-server 2 10.38.1.6))))))</pre>
	
Note	(create-v6-vendor-option <i>opt val</i>) is a synonym for (create-vendor-option) and can be used instead.

**Note**

We recommend that you use `create-option` and `create-vendor-option` for v4 and v6.

Creating and Editing Prefixes

You can create prefixes directly (and optionally apply an existing template to it; see the “[Creating and Editing Prefix Templates](#)” section on page 28-20). These are the prefix attributes that you can set:

- *name*—Assigns a name to this prefix.
- *vpn-id*—VPN that contains the prefix.

- **description**—Describes the prefix.
- **dhcp-type**—Defines how DHCP manages address assignment for a prefix:
 - **dhcp** (preset value)—Uses the prefix for stateful address assignment.
 - **stateless**—Uses the prefix for stateless option configuration.
 - **prefix-delegation**—Uses the prefix for prefix delegation.
 - **infrastructure**—Uses the prefix to map a client address to a link, when the prefix does not have an address pool.
 - **parent**—Do not have DHCP use the prefix. But, use it as a container object to group child prefixes. Parent prefixes appear only in the IPv6 address space listing in the web UI, not in the prefixes listing.
- **address**—Prefix (subnet) to which an interface belongs to, using the high-order bits of an IPv6 address.
- **owner**—Owner of the prefix.
- **region**—Region for the prefix.
- **reverse-zone-prefix-length**—Prefix length of the reverse zone for ip6.arpa updates. (See the “[Determining Reverse Zones for DNS Updates](#)” section on page 30-4 for details.)
- **range**—Subrange the server can use to configure prefixes for address assignment. The prefix used depends on the value set for the *dhcp-type* attribute. If unset, the prefix address applies. This value can specify a longer prefix than the prefix address to limit the range of addresses or prefixes available for assignment. (See the “[Links and Prefixes](#)” section on page 28-2 for details.)
- **link**—Link associated with the prefix (subnet), used to group prefixes that are on a single link.
- **policy**—Shared policy to use when replying to clients.
- **selection-tags**—List of selection tags associated with the prefix.
- **allocation-algorithms**—One or more algorithms the server uses to select a new address or prefix to lease to a client. The available algorithms are:
 - **client-request** (preset to off)—Controls whether the server uses a client requested lease.
 - **reservation** (preset to on)—Controls whether the server uses an available reservation for the client.
 - **extension** (preset to on)—Controls whether the server calls extensions attached at the **generate-lease** extension point to generate an address or prefix for the client. When you use generate-lease extension point with DHCPv6 failover, the server uses the address or delegated prefix that the extension returns and does not perform a hash on this address or prefix as it does with the randomly generated addresses. If the extension is using some algorithmic method to generate the address or delegated prefix then the extension must be failover aware (extension will be able to determine if failover configuration is enabled and the role of the failover server). For details on extensions, see [Chapter 31, “Using Extension Points.”](#)
 - **interface-identifier** (preset to off)—Controls whether the server uses the interface-identifier from the client (link-local) address to generate an address; ignored for temporary addresses and prefix delegation.
 - **random** (preset to on)—Controls whether the server generates an address using an RFC 3041 algorithm; ignored for prefix delegation.
 - **best-fit** (preset to on)—Controls whether the server delegates the first, best-fit available prefix; ignored for addresses.

When the server needs an address to assign to a client, it processes the flags in the following order until it finds a usable address: client-request, reservation, extension, interface-identifier, and random. When the server needs to delegate a prefix to a client, it processes the flags in the following order until it finds a usable prefix: client-request, reservation, extension, and best-fit.

- ***restrict-to-reservations***—Controls whether the prefix is restricted to client (or lease) reservations.
- ***max-leases***—Maximum number of nonreserved leases allowed on the prefix. When a new lease needs to be created, the server does so only if the limit is not exceeded. When the limit is exceeded, the server cannot create or offer new leases to clients. If you also enable SNMP traps, the *max-leases* value also calculates the percentage of used and available addresses.

**Tip**

Set the *max-leases* value to the expected maximum so that the SNMP address traps can return meaningful results.

- ***ignore-declines***—Controls whether the server responds to a DHCPv6 DECLINE message that refers to an IPv6 address or a delegated prefix from this prefix. If enabled, the server ignores all declines for leases in this prefix. If disabled (the preset value) or unset, the server sets to UNAVAILABLE every address or delegated prefix requested in a DECLINE message if it is leased to the client.
- ***expiration-time***—Time and date at which a prefix expires. After this date and time, the server neither grants new leases nor renews existing leases from this prefix. Enter a value in the format "[weekday] month day hh:mm[:ss] year"; for example, "**Dec 31 23:59 2006**". See the explanation for *expiration-time* attribute under “[Creating and Editing Prefix Templates](#)” section on page 28-20.
- ***free-address-config***—Identifies which trap captures unexpected free address events on this prefix. If not configured, the server looks for the *free-address-config* attribute value for the parent link. If that attribute is not configured, the server looks at its *v6-default-free-address-config* attribute.
- ***deactivated***—Controls whether a prefix extends leases to clients. A deactivated prefix does not extend leases to any clients and treats all addresses in its ranges as if they were individually deactivated. The preset value is false (activated).
- ***max-pd-balancing-length***—Controls the maximum prefix-delegation prefix length that the failover pool balancing will consider in balancing a prefix-delegation prefix. The default value is 64 and it should never be longer than the longest prefix length allowed for the prefix delegation.
- ***allocation-group***—Allocation group to which this prefix belongs.
- ***allocation-group-priority***—Priority of this prefix over other prefixes in the same allocation group. The default value is zero.
- ***embedded-policy***—Policy embedded in the prefix.

Local Advanced and Regional Web UI

-
- Step 1** From the Design menu, choose **Prefixes** under the **DHCPv6** submenu. The List/Add DHCP v6 Prefixes page shows the existing prefixes.
- Step 2** To create the prefix:
- If creating it in other than the current VPN, choose a VPN from the *username* drop-down list on the top right of the window at the top right of the window.
 - Click the **Add Prefixes** icon in the Prefixes pane, enter a prefix name and address, and choose a prefix length from the drop-down list.
 - If you want a range of addresses for the prefix, enter the subnet address and choose a prefix length.

- d. Choose a DHCP type (see the attribute descriptions at the top of this section). The default is DHCP.
- e. If you want to apply a preconfigured prefix template, choose it from the drop-down list. (Note that the attribute values of an applied template overwrite the ones set for the prefix.)
- f. Click **Add IPv6 Prefix**, which should add the prefix to the list.
- g. Reload the DHCP server. When you return to the List/Add DHCPv6 Prefixes page, a message indicates how many prefixes are synchronized.

Step 3 To create a reverse zone from the prefix, click the Reverse Zone tab. On this tab, you can select a zone template, and click **Report**, then **Run**.

Step 4 Once you create a prefix, you can view and manage the leases for the prefix by clicking the Leases tab. On the Leases tab, you can view the leases for the client lookup key and manage each lease separately by clicking its name.

Step 5 You can view and manage the reservations for the prefix by clicking the Reservations tab. Add each reservation IP address and lookup key and whether the lookup key is a string or binary, then click **Add Reservation**.

Step 6 To edit a prefix, click its name on the Prefixes pane. On the Edit Prefix page, edit the prefix attributes, assign prefix to a group and set priorities, or create a new or edit an existing embedded policy.

To assign the prefix to a group and set priorities:

- a. Enter the name of the group in the allocation-group attribute field.
- b. Enter the priority value in the allocation-group-priority attribute field. If you do not enter any value here, it will be allotted the default value (0) and this prefix will have the lowest priority in the group.

You can find these attributes under Allocation Group in Advanced mode (see [“Prefix Allocation Groups” section on page 28-6](#)).

To manage an embedded policy:

- a. Click **Create New Embedded Policy** or **Edit Existing Embedded Policy** to open the Edit DHCP Embedded Policy for Prefix page.
- b. Modify the embedded policy properties (see the [“DHCPv6 Policy Hierarchy” section on page 28-12](#)).
- c. Click **Modify Embedded Policy**. The next time the Edit DHCPv6 Prefix page appears, you can edit the embedded policy for the prefix.
- d. Click **Save**.

Step 7 In the regional web UI, you can push prefixes to local clusters and reclaim prefixes on the List/Add DHCPv6 Prefixes page:

- To push the prefix, click **Push** to open the DHCPv6 Push Prefix page. Choose the cluster or prefix template to which you want to push the prefix, then click **Push Prefix**. When the prefix is pushed, the reservations on the prefix is pushed with the prefix. Also, if the prefix is on a link, the parent prefix is pushed if it is not already present on the local cluster.
- To reclaim the prefix, click **Reclaim** to open the DHCPv6 Reclaim Prefix page. Choose the cluster or prefix template to which you want to reclaim the prefix, then click **Reclaim Prefix**. When the prefix is reclaimed, the reservations are deleted with the prefix, if there are no active leases, or if the force option is specified. Otherwise the prefix is deactivated.

**Note**

If the prefix is on a universal link, it can be pushed to more than one cluster and that local changes will not take effect until the next server reload.

CLI Commands

Use **prefix name create ipv6address/length**. (The **prefix** command is a synonym for the **dhcp-prefix** command from previous releases.) Reload the DHCP server. For example:

```
nrcmd> prefix example-prefix create 2001:0db8::/32 [attribute=value]
nrcmd> dhcp reload
```

To apply a prefix template during prefix creation, use **prefix name create ipv6address/length template=name**. To apply a template to an existing prefix definition, use **prefix name applyTemplate template-name**. For example:

```
nrcmd> prefix example-prefix create 2001:0db8::/64 template=preftemp-1
nrcmd> prefix example-prefix applyTemplate template=preftemp-1
nrcmd> dhcp reload
```

You can set and enable the aforementioned attributes in the usual way. Add reservations by using **prefix name addReservation ipv6address/length lookup-key [-blob | -string]**. List leases by using **prefix name listLeases**.

**Tip**

See the [“Reconfigure Support” section on page 28-35](#) for additional syntax.

You can get an exact count of the total prefixes and links for the DHCP server by using **dhcp getPrefixCount [vpn name | all]**. You can specify a VPN or all VPNs. Omitting the **vpn name** returns a count for the current VPN.

Viewing IPv6 Leases

You can view the leases for prefixes.

Local Basic or Advanced Web UI

To view leases, choose **Prefix** from the **Design** menu to open the List/Add DHCP v6 Prefixes page, then click the Leases tab for the prefix. This opens the page, where you can click each lease to manage it.

See the [“IPv6 Lease States” section on page 28-8](#) for a description of the values in the State column. For guidelines as to the lease expiration time, see the [“Guidelines for Lease Times” section on page 24-3](#).

To manage the DHCP leases, click the lease IP address.

CLI Commands

Use **lease6 ipaddr show** to show the properties of a particular lease based on its IP address. To manage DHCPv6 leases use the following 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
```

**Note**

In Cisco Prime Network Registrar 8.2, the **lease6 list** command lists available and other-available prefix delegation leases that are not bound to a client.

Viewing Address Utilization for Prefixes

You can view the current address utilization for prefixes.

Local Advanced and Regional Web UI

The function is available on the DHCP v6 Address Tree page (see the [“Viewing Address Space” section on page 9-3](#)).

**Tip**

You can use the DHCP v6 Address Tree page to push and reclaim prefixes. Click the **Push** or **Reclaim** icon for the desired prefix. (See in the [“Creating and Editing Prefixes” section on page 28-25](#) for details.)

When you click the Current Usage tab, the Utilization Detail appears.

**Note**

To ensure the proper subnet-to-server mapping on this page, you must update the regional address space view so that it is consistent with the relevant local cluster. Do this by pulling the replica address space, or reclaiming the subnet to push to the DHCP server. Also ensure that the particular DHCP server is running.

The other columns under the Current Usage tab identify:

- **Range**—Address range of the prefix.
- **Type**—Whether the address space is a prefix or link.
- **Active Dynamic**—Addresses that are part of a dynamic range managed by DHCP and that are currently leased, but not reserved.
- **Allocation Group**—Allocation group to which the prefix belongs.

The items under Utilization Detail are expandable so that you can view the prefix or parent prefix data.

The Utilization Detail page is a read-only page that shows detailed address utilization attributes for the prefix or the parent prefix (identified as Totals). The address utilization attributes are described in [Table 28-6](#).

Table 28-6 Address Utilization Attributes

Utilization Attribute	Description
<i>Tenant</i>	
<i>aggregation-level</i>	Granularity of the utilization data. Prefix-level indicates the data is for the specific prefix; totals indicates the data is for the parent prefix, which is the sum of its prefix-level counters.
<i>dhcp-type</i>	DHCP address assignment type, which can be dhcp (stateful), stateless (option configuration), prefix-delegation, or infrastructure (maps a client address to a link without an address pool).
Total Addresses	
<i>active-dynamic</i>	Total number of dynamic leases in active use (leased, offered, released, expired, or revoked). The Active Dynamic category shows the states of these leases.
<i>total-reserved</i>	Total number of reserved leases.
Active Dynamic	
<i>offered</i>	Number of dynamic (unreserved) leases that are currently offered to clients, but not yet acknowledged as being leased.
<i>leased</i>	Number of dynamic leases that are currently acknowledged as leased to clients.
<i>expired</i>	Number of dynamic leases that are past the lease expiration period, but will not be available for other clients (except after the policy grace-period expires).
<i>revoked</i>	Number of dynamic leases that the client can no longer use, but that some other client could be using.
Reserved	
<i>reserved-active</i>	Number of reserved leases that clients are actively using.
<i>reserved-inactive</i>	Number of reserved leases that clients are not actively using.
Unavailable	
<i>unavail</i>	Number of unreserved dynamic leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).
<i>reserved-unavail</i>	Number of reserved leases that a client declines or the server marks with an address conflict (usually indicating configurations that need correcting).
Deactivated	
<i>deactivated</i>	Number of dynamic and reserved leases that clients are actively leasing (that are not offered, expired, or released), but that an administrator deactivated.
<i>leased-deactivated</i>	Number of dynamic leases that an administrator deactivated.
<i>reserved-leased-deactivated</i>	Number of reserved leases that an administrator deactivated.
Prefix Delegation Lease Counts	
<i>max-pd-balancing-length</i>	Prefix length used for counting the prefix delegation leases.
<i>prefixes-in-use</i>	Number of prefixes, of the max-pd-balancing-length prefix length, that are in use.
<i>prefixes-available</i>	Number of prefixes, of the max-pd-balancing-length prefix length, available to any client on this server.

Table 28-6 Address Utilization Attributes (continued)

Utilization Attribute	Description
<i>prefixes-other-available</i>	Number of prefixes, of the max-pd-balancing-length prefix length, available to any client on the failover partner.
<i>prefixes-in-transition</i>	Number of prefixes, of the max-pd-balancing-length prefix length, in transition between the failover partners.
Failover Related	
<i>available</i>	Number of prefix delegation leases available to any client on this server. This is the count of the number of lease objects and not the number of prefixes of a specific prefix-length.
<i>other-available</i>	Number of prefix delegation leases that this server believes the partner has available to any client.
<i>pending-available</i>	Number of leases that are in the pending-available state.
<i>pending-delete</i>	Number of leases that are in the pending-delete state.

Viewing DHCPv6 Networks

To view the networks in the DHCPv6 address space, choose **Networks** from the **Design > DHCPv6** menu, to open the DHCPv6 Network Tree page. On this page you can add DHCPv6 links using a template and a template root prefix, as you would on the List/Add DHCPv6 Links page. Adding a link opens the Add DHCPv6 Link page. After creating the link, you can select it on the View DHCPv6 Networks page for editing.



Tip

You can use the DHCP v6 Network Tree page to push and reclaim links. Click the **Push** or **Reclaim** icon for the desired link. (See in the “[Creating and Editing Links](#)” section on page 28-18 for details.)

Editing DHCPv6 Server Attributes

You can edit DHCP server attributes related to DHCPv6. These attributes are:

- ***v6-client-class-lookup-id***—Expression that determines a client-class based on the DHCPv6 client request and returns a string with either the name of a configured client-class or <none> (if the expression does not wish to provide a client-class). The attribute has no preset value.
- ***max-client-leases***—Maximum number of leases a DHCPv6 client can have on a link. Do not use this attribute to limit clients to one lease only. The preset is 50.

Local Basic or Advanced Web UI

From the **Deploy** menu, choose **DHCP Server** under the **DHCP** submenu to open the Manage DHCP Server page. Click the Local DHCP Server link to open the Edit DHCP Server page, modify the aforementioned DHCPv6 attribute values, then click **Save**.

CLI Commands

Use **dhcp** to show the aforementioned DHCPv6 server attributes, then modify them by using **dhcp set**.

Configuring DHCPv6 Policies

You can edit DHCPv6 policy attributes, which are:

- *affinity-period*—See the “[Lease Affinity](#)” section on page 28-8 (no preset value).
- *allow-non-temporary-addresses*—Enable or disable DHCPv6 clients requesting nontemporary (IA_NA) addresses (preset value enable).
- *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 (preset value 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 28-12 for special handling of this attribute, and Reconfigure support, when used in an embedded or named policy for a prefix.)
- *allow-temporary-addresses*—Enable or disable DHCPv6 clients requesting temporary (IA_IA) addresses (preset value enable).
- *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 prefix length (preset value 64 bytes).
- *preferred-lifetime*—Default and maximum preferred lifetime for leases (preset value 1 week).
- *v6-reply-options*—DHCPv6 options returned in replies to clients (no preset value). (See the “[DHCPv6 Policy Hierarchy](#)” section on page 28-12 for special handling of this attribute when used in an embedded or named policy for a prefix.)
- *valid-lifetime*—Default and maximum valid lifetime for leases (preset value 2 weeks).

**Tip**

For details on the Reconfigure attributes, see the “[Reconfigure Support](#)” section on page 28-35.

Local Advanced Web UI

- Step 1** From the **Design** menu, choose **Policies** under the **DHCP Settings** submenu to open the List/Add DHCP Policies page.
- Step 2** Click the **Add Policy** icon in the Policies pane to add a new policy or select an existing policy to open the Edit DHCP Policy page.
- Step 3** The DHCP Policies page has DHCPv4 and DHCPv6 options sections. Add (or delete) options and set attributes as desired, then click **Save**.

CLI Commands

Use **policy list** or **policy name show** to show the aforementioned policy attributes, then modify them by using **policy name set** or **enable**.

Configuring DHCPv6 Client-Classes

You can configure DHCPv6 client-class attributes, which are:

- ***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.

Local Advanced Web UI

-
- Step 1** From the **Design** menu, choose **Client Classes** under the **DHCP Settings** to open the List/Add DHCP Client Classes page.
- Step 2** Click an existing client-class to open the Edit DHCP Client Class page, or click the **Add Client Classes** icon in the Client Classes pane to add a new client-class on the Add DHCP Client Class dialog box. Both pages include the aforementioned attributes.
- Step 3** Click **Save**.
- Step 4** To generate clients, be sure that *validate-client-name-as-mac* is disabled for the DHCP server. This attribute appears on the Edit DHCP Server page under the Client-Class attributes.
- Step 5** Reload the DHCP server.
-

CLI Commands

Use **client-class list** or **client-class name show** to show the aforementioned client-class attributes, then modify them using **client-class name set**. To generate clients, be sure that *validate-client-name-as-mac* is disabled for the DHCP server.

Configuring DHCPv6 Clients

You can configure DHCPv6 clients.

Local Advanced Web UI

From the **Design** menu, choose **Clients** under the **DHCP Settings** submenu to open the List/Add DHCP Clients page. Select an existing client to open the Edit DHCP Client page or click the **Add Clients** icon on Clients pane to add a new client-class on the List/Add DHCP Client page, choose the client-class that includes the DHCPv6 attributes that were set (see the “[Configuring DHCPv6 Client-Classes](#)” section on [page 28-34](#)), then click **Save**.



Tip

Disable the *validate-client-name-as-mac* attribute for the DHCP server.

CLI Commands

Use **client list** or **client name show** to show the existing clients. To set the client-class name for the client, use **client name set client-class-name=value**. Also ensure that the *validate-client-name-as-mac* attribute is disabled for the DHCP server.

Setting DHCPv6 Options

Set DHCPv6 options and vendor options when you create or edit policies (embedded or named) for prefixes. (See the “[DHCPv6 Policy Hierarchy](#)” section on page 28-12 for special handling of the *v6-options* and *v6-vendor-options* policy attributes when used in an embedded or named policy on a prefix.)

Local Advanced Web UI

The DHCPv6 options coexist along with the DHCPv4 options on the List/Add DHCP Policies 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 23-8).

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.

CLI Commands

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 foo setV6VendorOption 17 dhcp6-cablelabs-config "(32 2222::3,2222::4)"
```

Reconfigure Support

For DHCPv6, a server can send a RECONFIGURE message to a client to inform the client that the server has new or updated configuration parameters. If so authorized and through proper authentication, the client then immediately initiates a Renew, Rebind, or Information-request reply transaction with the server so that the client can retrieve the new data. Without this support, a client must wait until it renews its lease to get configuration updates.

You can have the server unicast the Reconfigure packet or deliver it through a relay agent. If you do not specify either way, the client’s client-class policy, requested lease’s prefix or link policies, or system_default_policy (but not the client policy) determines the preferred method. If the unicast method is not available (the client has no valid address lease), the server uses the relay agent; with no relay agent, the server tries to unicast; failing both results in an error. With the unicast method, if the specified lease is not usable, the server selects the lease with the longest valid lifetime.

The server and client negotiate Reconfigure support through the added security of a reconfigure key. The internal process is basically:

1. The client sends the server a REQUEST, SOLICIT, or ADVERTISE packet that includes the *reconfigure-accept* option (20) to indicate that the client wants to accept Reconfigure messages. (Conversely, the DHCP server can send a *reconfigure-accept* option to the client about whether the client should accept Reconfigure messages.) This option is required for Reconfigure support.
2. If the Cisco Prime Network Registrar policy for the client has the *reconfigure* attribute set to **allow** or **require** (rather than **disallow**), the DHCP server accepts the packet and generates a reconfigure key for the client. (The server records the key value and its generation time in the *client-reconfigure-key* and *client-reconfigure-key-generation-time* attributes for the DHCPv6 lease.)
3. The server sends a Reply packet to the client with the reconfigure key in the *auth* option (11) along with the *reconfigure-accept* option.
4. The client records the reconfigure key to authenticate Reconfigure messages from the server.

5. When the server wants to reconfigure the client, it sends a Reconfigure packet with the *reconfigure-message* option (19) and an *auth* option containing a hash generated from the packet and the reconfigure key. The *reconfigure-message* option indicates in the *msg-type* field whether the client should respond with a Renew or an Information-request packet.
6. Upon receiving the packet, the client validates that the *auth* option contains the valid hash, then returns a Renew, Rebind, or Information-request packet. This packet includes an Option Request (*oro*) option (6) to indicate specific option updates. (If the server does not receive a reply from the client in a preconfigured timeout value of 2 seconds, the server retransmits the Reconfigure message at most 8 times, then aborts the reconfigure process for the client.)
7. The server sends the client a Reply packet that includes options for configuration parameters. The packet might also include options containing addresses and new values for other configuration parameters, even if the client did not request them. The client records these changes.

**Note**

For details on how Reconfigure support affects particular DHCP extension points, see the “[Extension Dictionaries](#)” section on page 31-23.

Local Advanced Web UI

The List/Add DHCP Leases for Prefix page includes a **Reconfigure** button for each lease so that you can initiate a reconfiguration request for that particular lease.

CLI Commands

To support Reconfigure, Cisco Prime Network Registrar includes the following syntax for the **lease6** command:

```
lease6 ipaddr reconfigure [renew | rebind | information-request] [-unicast | -via-relay]
```

The options determine whether to have the client respond to the Reconfigure message with a Renew, Rebind, or Information-request packet, and whether the server should unicast or go through a relay agent. The **lease6 list** and **show** commands also display values for these related attributes:

- **client-reconfigure-key**—128-bit key that the server generates for Reconfigure messages to the client.
- **client-reconfigure-key-generation-time**—Time at which the server generated the *client-reconfigure-key*.

The **policy** command includes two related attribute settings:

- **reconfigure**—Whether to allow (1), disallow (2), or require (3) Reconfigure support; the preset value is allow (1).
- **reconfigure-via-relay**—Whether to allow reconfiguration over a relay agent; the preset value is false, whereby reconfiguration notification is by unicasting from the server.

DNS Update for DHCPv6

For details on enabling and configuring DNS update for DHCPv6 clients, see the “[DNS Update for DHCPv6](#)” section on page 30-2.