Managing Leases

Leases are at the center of the Dynamic Host Configuration Protocol (DHCP). They are the IP addresses allocated to individual clients for a certain time period. The DHCP server automatically allocates these leases with properly configured scopes that include valid IP address ranges. No two clients can have the same leased address. Reservations are leases that always get the same IP address.

This chapter describes how to manage leases and reservations in a network.

- Lease States, on page 1
- Guidelines for Lease Times, on page 2
- DHCPv6 Clients and Leases, on page 5
- Configuring Leases in Scopes, on page 7
- Viewing Leases, on page 7
- Using Client Reservations, on page 15
- Creating Lease Reservations, on page 18
- Setting Advanced Lease and Reservation Properties, on page 23
- Querying Leases, on page 32
- Running Address and Lease Reports, on page 38
- Dynamic Lease Notification, on page 46
- Sample Lease Notification Client, on page 48
- Lease History Database Compression Utility , on page 53
- Elastic Lease Times, on page 58

Lease States

The tables below list the IPv4 or IPV6 lease states.

IPv4 Lease States

A IPv4 lease can be in one of the states described in the table below.

Table 1: IPv4 Lease States

State	Description
Available	IP address available to be leased.

State	Description
Unavailable	Not leasable. See Handling Leases Marked as Unavailable, on page 31 for ways the DHCP server might set a lease to unavailable.
Leased	Held by a client.
Offered	Offered to the client.
Expired	Available when the lease grace period expires.
Deactivated	Not renewable or leasable after the lease expires. See Deactivating Leases, on page 10.
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. See Managing DHCP Failover.

IPv6 Lease States

A lease can be in one of the states described in the table below.

Table 2: 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.
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.

Guidelines for Lease Times

To define appropriate values for lease times, consider these events on your network:

- Frequency of changes to DHCP options and default values.

- Number of available IP addresses compared to clients requesting them.
- Number of network interface failures.
- Frequency at which computers are added to and removed from the network.
- Frequency of subnet changes by users.

All these events can cause clients to release IP addresses or the leases to expire at the DHCP server. Consequently, the addresses may return to the free-address pool for reuse. If many changes occur on your network, Cisco recommends a lease time between one and three days for active networks, and between four and ten days for inactive networks. Assigning such a lease time reassigns IP addresses more quickly as clients leave the subnet.

Another important factor is the ratio of available addresses to connected computers. For example, the demand for reusing addresses is low in a class C network having 254 available addresses, of which only 40 are used. A long lease time, such as two months, might be appropriate in such a situation. The demand would be much higher if there were 240 to 260 clients trying to connect at one time. In this situation, you should try to configure more address space. Until you do, keep the DHCP lease time to under a hour.

Tip Short lease periods increase the demand that the DHCP server be continuously available, because clients will be renewing their leases more frequently. The DHCP failover functionality can help guarantee such levels of availability.

Be careful when creating policies that have permanent leases. A certain amount of turnover among clients occurs, even in a stable environment. Portable hosts might be added and removed, desktop hosts moved, and network adapter cards replaced. If you remove a client with a permanent lease, it requires manual intervention in the server configuration to reclaim the IP address. It would be better to create a long lease, such as six months, to ensure that addresses are ultimately recovered without administrator intervention.

Recommendations for lease durations include:

- Set cable modem lease times to seven days (604800 seconds). The leases should come from private address space, and the cable modems should seldom move around.
- Leases for customer premises equipment (CPE) or laptops should come from public address space and should match the habits of the user population, with as long a lease as possible to reduce load on the server.
- Shorter lease times require more DHCP request and response buffers. Set the request and response buffers for optimal throughput (see Setting DHCP Request and Response Packet Buffers).
- Allow the server to determine the lease period, by ensuring that the *allow-lease-time-override* policy attribute is disabled, which is its normal default. Even if enabled, clients can only request lease times that are shorter than you configure for the server. Some clients always request a fixed lease time (such as an hour) or the same one they had previously. These kinds of requests can cause problems in that the client never gets the full lease time, thereby generating more traffic for the server.
- Defer any lease extensions for clients trying to renew leases before the halfway mark in the lease. For details, see Deferring Lease Extensions.

Restricting Lease Dates

Lease date restrictions can be specified using the following attributes:

- *lease-retention-max-age*
- *lease-retention-min-age*

The *lease-retention-max-age* attribute specifies the longest time, in the past (from the current time), to which lease times are restricted. This can be used to meet data retention restrictions for privacy protection. If not specified, no restrictions are placed on how far back in time the lease times may be. In order for lease retention limitation to take place for a lease, not only does the *lease-retention-max-age* need to be non-zero, but the individual lease itself must fall under a policy where the *lease-retention-limit* attribute is set in that policy. This value, if configured, must be greater than 8 hours. If it is configured as non-zero and less than eight hours, it will be set to eight hours.

The *lease-retention-min-age* attribute specifies the shortest time, in the past, to which lease times may be restricted. Its value must be at least 6 hours less than the *lease-retention-max-age*. If this attribute is enabled and is configured to a non-zero value, lease times subject to retention limitation will not be allowed to grow older than *lease-retention-max-age*. As they progress toward *lease-retention-max-age*, they are periodically reset to *lease-retention-min-age* in the past. Configuring this attribute is optional as it will be six hours less than the *lease-retention-max-age,* by default. Also if the difference between the attribute values is less than six hours then *lease-retention-max-age* minus six hours is used.

Keeping older times on a lease between *lease-retention-min-age* and *lease-retention-max-age* involves some processing, and the closer these two values are, the more frequently this processing must take place, regardless of the absolute values of these attributes. Setting the *lease-retention-min-age* to several days before the *lease-retention-max-age* minimizes the additional server processing devoted to lease retention limitation.

You have to change one or more policies for the clients which are subject to these retention times. You can configure this in the system_default_policy to apply to all clients. But if there are some devices for which this does not matter, it might be best to configure it more selectively. The fewer the clients with this feature enabled, the lesser the impact on the performance of the server because of lesser work.

The policy attribute *lease-retention-limit* indicates whether the clients associated with that policy are subject to the lease date restrictions. If this attribute is enabled and the *lease-retention-max-age* of the DHCP server is configured to a non-zero value, lease times subject to this policy will not be allowed to grow older than *lease-retention-max-age*. As they progress toward *lease-retention-max-age*, they will periodically be reset to lease-retention-min-age in the past.

Some points to remember when considering to use the privacy protection feature are:

- When first enabled (or for certain reconfigurations), existing lease history records will not be subject to this feature because these records will not have the *lease-retention-limit* flag set.
- The lease history trimming time will likely be adjusted. It is set to about two-thirds of the difference between the *lease-retention-max-age* and *lease-retention-min-age* values. For example, when the default value of six hours is taken, the trimming is done every 4 hours.
- Disk Input/Output rates go up on the system. This is because the server needs to update the older times in the active and historical lease records. The impact of this can be reduced to some extent by increasing the difference between the *lease-retention-max-age* and *lease-retention-min-age* values.
- When configuration changes such as removing scopes and prefixes, or adjusting ranges are made, the leases associated with the scopes or prefixes become orphaned leases. These orphaned leases are not trimmed and not processed for privacy protection time limits. You must remove the orphaned leases. For more information, see Removing Orphaned Leases, on page 12.

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).
- 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 8415).

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 then 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.

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).

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:
   a. Transitions to DELETE state and the server deletes it from memory and the lease database after the affinity-period elapses.

   b. 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.

Configuring Leases in Scopes

After setting the IP address ranges for a scope, you can monitor and adjust the leases that result from DHCP assignments.

Viewing Leases

To view leases, you must first create a range of IP addresses for them in a scope, as described in the "Set Up DHCP" chapter of the *Cisco Prime Network Registrar 11.2 Quick Start Guide* or the Managing Scopes, then wait for the DHCP server to generate leases based on these addresses.

Local Basic Web UI

From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page, then click the **Leases** tab for the scope. This opens the page, where you can click each lease to manage it.

See Lease States, on page 1 for a description of the values in the State column. For guidelines as to the lease expiration time, see Guidelines for Lease Times, on page 2.

To open the Edit DHCP Scope page, click the lease IP address.

Local Advanced Web UI

From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page. You can then click the **Leases** tab for the scope; or you can click the name of the scope to open the Edit DHCP Scope page, then click **Leases** tab in the page.

CLI Commands

Use **lease** [*vpn-name*/]*ipaddr* **show** to show the properties of a particular lease based on its IP address. Use **scope** *name* **listLeases** to show all the leases for a named scope. The output is nearly identical for both commands. Note that you cannot list leases in a particular virtual private network (VPN); all the leases in all the VPNs appear in the list.

You can show the most recent MAC address associated with a lease or what lease is associated with a MAC address. The **lease** [*vpn-name*/]*addr* **macaddr** command shows the MAC address of the lease, whether or not that lease is reserved or active. The **lease list –macaddr** *addr* [**-vpn**=*vpn-name*] command lists the lease data only if the IP address for that MAC address was actively leased (and not reserved). You can also list leases by LAN segment and subnet by using **lease list -lansegment** *addr mask* and **lease list -subnet** *addr mask* commands.

Importing and Exporting Lease Data

You can use the CLI to import lease data to, and export from, text files.

Import Prerequisites

Before you can import leases, you must perform several configuration steps:

1. Configure a scope or scopes in the DHCP server for the leases that you plan to import.
2. If you want the hostnames for the leases dynamically entered into DNS as part of the import, configure zones in the DNS server to allow dynamic updates from the DHCP server.
3. Set the DHCP server to import mode so that it does not respond to other lease requests during the lease importing.
4. For all the time fields, use either the number of seconds since midnight GMT January 1, 1970, or a day, month, date, time, year format (Mon Apr 15 16:35:48 2002).
5. After you import the leases, take the DHCP server out of import mode so that it can respond to other lease requests.

Note Importing permanent leases will fail if you disable the permanent leases option. Enable this option using **policy** *name* **enable permanent-leases**, as necessary.

Import and Export Commands

The **import leases** and **export leases** commands use a special file format. Each record, or line, in the file represents one DHCP client:

```
field-1|field-2|field-3|...|field-13
```

Do not use spaces between the vertical line (|) delimiter and the field values. You must include at least the first four required fields. If you include more, you must delimit all the remaining null fields with the vertical line (|) so that there are 13 fields. The fields are, in order:

1. MAC address in *aa*:*bb*:*cc*:*dd*:*ee*:*ff* format (required)
2. MAC address type (required)
3. MAC address length (required)
4. IP address in dotted decimal format, *a*.*b*.*c*.*d* (required)
5. Start of lease time (Greenwich Mean Time, GMT) (optional)
6. Lease expiration time (GMT) (optional)
7. Allowable extension time (GMT) (optional)
8. Last transaction time (GMT) (optional)
9. IP address of the DHCP server (optional)
10. Hostname (without domain) (optional)

11. Domain name (optional)
12. Client ID (optional)
13. VPN name (optional; if omitted, the global VPN is used)

For all the time fields, use either the number of seconds since 1970, or the *day-month-date-time-year* format (such as Mon Apr 9 16:35:48 2007).

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

When you use **export leases**, you can choose between writing the state of all current and expired leases, or just the current leases, to the output file. The example below shows part of a lease data export from a Cisco Prime Network Registrar DHCP server. The blank lines between records appear in the example for clarity; they are not in the actual output.

Example: Lease Data Export

```
00:60:97:40:c1:96|1|6|204.253.96.103|Wed Aug 30 08:36:57 2000|Fri Sep 01 13:34:05 2000|
Wed Aug 30 08:36:57 2000|Fri Sep 01 09:34:05 2000|204.253.96.57|nomad|cisco.com|
00:d0:ba:d3:bd:3b|blue-vpn
00:d0:ba:d3:bd:3b|1|6|204.253.96.77|Thu Aug 17 13:10:11 2000|Fri Sep 01 14:24:46 2000|
Thu Aug 17 13:10:11 2000|Fri Sep 01 10:09:46 2000|
204.253.96.57|NPI9F6AF8|cisco.com|blue-vpn
00:d0:ba:d3:bd:3b|1|6|204.253.96.78|Fri Jun 23 15:02:18 2000|Fri Sep 01 14:11:40 2000|
Fri Jun 23 15:02:18 2000|Fri Sep 01 09:56:40 2000|
204.253.96.57|JTB-LOCAL|cisco.com|blue-vpn
```

Lease Times in Import Files

For a lease import request, if the DHCP server is:

- Enabled for *import-mode* and the lease is not already leased to the client, the server accepts any lease time the client specifies.
- Enabled for *import-mode*, the lease is already leased to the client, *defer-lease-extensions* is enabled for the server (the default), and the request arrives before the renewal time (T1), the server uses the existing lease time.

  If the request arrives after T1, the server gives the client whatever it asks for. Within about two minutes of the expiration time, *defer-lease-extensions* is inoperative.

- Not enabled for *import-mode*, it never accepts a lease time longer than the server-configured one.
  - If *allow-lease-time-override* is enabled for a policy applicable to the request, the server accepts a shorter lease time from the client. The shorter lease time is acceptable to the server, even though you can set a server expert mode *client-requested- min-lease-time* attribute that creates a floor for the lease time.
  - If *allow-lease-time-override* is not enabled for any applicable policy, the server ignores the dhcp-lease-time request in the incoming packet and uses the server setting.

If your import file specifies a DNS zone name, the server does not use the zone name when it updates the DNS. If the file specifies a hostname, then the server uses the hostname when updating the DNS, unless hostname specification in a client or client-class entry overrides the hostname.

The client hostname should be in a zone other than the zone associated with the DNS update configuration object used for the DNS update. This can be indicated to the DHCP server, only by specifying that zone in a client or client-class entry.

Pinging Hosts Before Offering Addresses

You can have the DHCP server use the Internet Control Message Protocol (ICMP) echo message capability (also known as **ping**) to see if anyone responds to an IP address, before assigning it (using the *ping-clients* attribute). The *ping-clients* attribute controls whether the server should attempt to ping an address before offering a lease. If enabled, then the *ping-timeout* attribute may also need to be set.This test allows the DHCP server to check whether an address is not in use before assigning it.

Using **ping** can help prevent two clients from using the same address. If a client responds to ping, the DHCP server marks that address as *unavailable* and offers a different address. This test works only for powered-up clients; it is possible for clients to have a lease and be powered down.

You can also configure the *ping-clients* attribute at the DHCP server. This attribute controls the default value of the *ping-clients* attribute of a scope, if not explicitly configured on a scope.

Note If you have configured scopes, the scope-specific configuration takes precedence; scopes without explicit configurations assume the global setting.

The ping timeout period is important. Because pinging helps to ensure that no client is using a particular IP address, each ping must wait the entire timeout period. This ping timeout period comes before an offer, so the time specified has a considerable effect on server performance.

- If you set this time too long, it slows down the lease offering process.
- If you set this time too short, it reduces the effectiveness of the ping packet to detect another client using the IP address.

To implement pinging hosts before offering IP addresses, modify the scope by:

- Enabling the *ping-clients* attribute. It is disabled by default.
- Setting the *ping-timeout* attribute. It is 300 milliseconds by default.

The server makes unavailable any IP address for which it receives a successful ECHO reply. You can control this action by enabling the DHCP server attribute *ignore-icmp-errors* (the preset value). If disabled, the DHCP server also uses ICMP DEST_UNREACHABLE and TTL_EXPIRED error messages that it receives after sending ICMP ECHO requests as reasons for making an IP address unavailable.

Deactivating Leases

Deactivating a lease moves a client off of it. If the lease is available, deactivating it prevents the DHCP server from giving it to a client. If the lease is active (held by a client), deactivating it prevents the client from

renewing it and the server from giving the lease to another client. You can deactivate a lease only if the server is running. The DHCP server deactivates the lease immediately.

Tip To force a Windows client to release its lease, run **ipconfig /release** on the client machine.

Note For DHCPv4 leases, the lease will remain deactivated until it is reactivated. For DHCPv6 leases (address or prefix delegation), the behavior is a bit different in that the lease will automatically be activated when the client is removed from the lease. Therefore, there is no need to activate DHCPv6 deactivated leases. However, this also means that the lease is available after the current lease ends and you cannot deactivate leases that are not associated with a client. If a DHCPv6 reservation is deactivated, it must be activated to be used by that client again.

Local Web UI

To deactivate a lease, click the address of the lease on the **Leases** tab for the Scope (see Viewing Leases, on page 7) and click **Deactivate**. The lease now shows as deactivated. To reactivate the lease, click **Activate**. You can also deactivate DHCPv6 leases in a similar manner.

CLI Commands

To deactivate a lease, use **lease** [*vpn-name*/]*ipaddr* **deactivate**. To reactivate a lease, use **lease** [*vpn-name*/]*ipaddr* **activate**.

To deactivate a DHCPv6 lease, use **lease6** [*vpn-name*/]*addr* **deactivate**. To reactivate a DHCPv6 lease, use **lease6** [*vpn-name*/]*addr* **activate** (though see the note above as DHCPv6 leases generally need not be re-activated as this happens automatically when the client is removed from the lease).

Excluding Leases from Ranges

IP address ranges, by definition, must be contiguous. To exclude a lease from an existing range, you must divide the range into two smaller ones. The new ranges consist of the addresses between the original starting and ending range addresses and the address that you want to exclude.

Caution If the excluded address currently has an active lease, you should first follow the steps in Deactivating Leases, on page 10, otherwise you will get a warning message. Deleting an active lease can result in a duplicate IP address if the deleted address is subsequently reconfigured and then reassigned. Information about that lease will no longer exist after you reload the server.

Local Basic Web UI

To exclude a lease from a scope address range, do the following:

Step 1 From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page.

Step 2 Click the name of the scope in the Scopes pane to open the Edit DHCP Scope page.

Step 3 In the Ranges area, click the **Delete** icon next to the IP address range you want to remove.

Step 4 Add a range that ends just before the excluded IP address.

Step 5 Add another range that begins just after the excluded IP address.

Step 6 Click **Save** to save the scope.

Step 7 Reload the DHCP server.

Local Advanced Web UI

To exclude a lease from a scope address range, the same operations exist as in Basic mode, except that you click the name of the scope on the List/Add DHCP Scopes page, which opens the Edit DHCP Scope page.

CLI Commands

To exclude a lease from a scope address range, discover the lease range (**scope** *name* **listRanges**), deactivate the lease (**lease** [*vpn-name*/]*ipaddr* **deactivate**), then remove the range of just that IP address (**scope** *name* **removeRange** *start end*). The resulting ranges are then split appropriately.

The following example removes the 192.168.1.55 address from the range. Note that if the lease is in a scope with a defined VPN, you must explicitly define that VPN for the session, or you can include the VPN prefix in the **lease** command:

```
nrcmd> session set current-vpn=red

nrcmd> scope examplescope1 listRanges

nrcmd> lease red/192.168.1.55 deactivate

nrcmd> scope examplescope1 removeRange 192.168.1.55 192.168.1.55

nrcmd> scope examplescope1 listRanges
```

Removing Orphaned Leases

To remove the orphaned leases:

Before you begin

When configuration changes such as removing scopes and prefixes, or adjusting ranges are made, the leases associated with the scopes or prefixes become orphaned leases. These orphaned leases are not updated periodically to assure that they do not violate the date restrictions.

When you use the lease date restriction feature, ensure that no orphaned-leases are present (or clean them out periodically).

Step 1 Enable the DHCP attribute *delete-orphaned-leases*:

```
nrcmd> dhcp enable delete-orphaned-leases
```

Step 2 Reload the DHCP server:

```
nrcmd> dhcp reload
```

Step 3 Unset the DHCP attribute *delete-orphaned-leases*:

```
nrcmd> dhcp unset delete-orphaned-leases
```

Step 4 Reload the DHCP server:

```
nrcmd> dhcp reload
```

Searching Server-Wide for Leases

Using Cisco Prime Network Registrar, you can search for leases, server-wide. The search is a filter mechanism whereby you can specify a combination of lease attributes to target one or more leases configured for the network. The lease history search function is available at both local and regional cluster whereas the active lease search function is available only at the local cluster. The search function is provided separately for DHCPv4 and DHCPv6 leases.

You can also search for the active leases using Cisco Prime Network Registrar.

Local Advanced Web UI

To search for DHCPv4 leases, do the following:

Step 1 From the **Operate** menu, choose **DHCPv4 Current Leases** under the **Reports** submenu to open the DHCP Lease Search page.

Note You can open the DHCP Lease Search page by clicking the Search button in the DHCP Lease History Search page (choose **DHCPv4 Lease History** under the **Reports** submenu to open the DHCP Lease History Search page). This button helps you to toggle between lease history search page and active leases search page.

Step 2 Choose a Filter Attribute from the drop-down list, such as address. DHCPv4 and DHCPv6 have separate lists of filter attributes. Also, the set of filter attributes are different for active and historical leases.

Attributes are greyed out after you select them as elements.

Step 3 Choose a filter Type from the drop-down list. You can choose at least Binary or Regular Expression, but the list can contain one or more of the following, depending on the Filter Attribute selected:

- Binary—Value is in binary notation.
- Date Range—Range of date values, From a date and time To a date and time.
- Integer—Value is an integer.
- Integer Range—Integer From value to an integer To value.
- IP Address—Value is an IP address.
- IP Range—IP address From value to an IP address To value.
- IP Subnet—Value is an IP subnet.
- Regular Expression—Value is a Regular Expression in regex syntax. (For common regex usage, see the *"Configuring Administrators" chapter in Cisco Prime Network Registrar 11.2 Administration Guide*.)

Step 4 Enter a Value, based on the Type selected. To clear the filter, click **Clear Filter**.

Step 5 Click **Add Element** to add the search element to the Filter Elements list. You can delete the element by expanding the filter display, then clicking the **Delete** icon next to the element.

Step 6 Once you assemble a list of elements, you can search on them, so that the elements are ANDed together for the result. Click **Search**.

Step 7 Check the table of resulting leases from the search, which shows for each an address, state, MAC address, hostname, flags, and expiration date. If necessary, change the page size to see more entries. The leases are ordered by IP address.

Tip The filter elements are ANDed together for the search. If you find that the search results do not yield what you expect, look at the Filter Elements list again and delete elements that can obstruct the results.

Local Advanced Web UI

To search for DHCPv6 leases, do the following:

Step 1 From the **Operate** menu, choose **DHCPv6 Current Leases** under the **Reports** submenu to open the DHCP v6 Lease Search page.

You can also go to the DHCP v6 Lease Search page if you choose **DHCPv6 Lease History** under the **Reports** submenu. If you choose **DHCPv6 Lease History** under the **Reports** submenu, the DHCP v6 Lease History Search page is displayed. You have to click the Search button to go to the DHCP v6 Lease Search page.

Step 2 Choose a Filter Attribute from the drop-down list, such as address.

Step 3 Choose a filter Type from the drop-down list. You can choose at least Binary or Regular Expression, but the list can contain one or more of the following, depending on the Filter Attribute selected:

- Binary—Value is in binary notation.
- Date Range—Range of date values, From a date and time To a date and time.
- Integer—Value is an integer.
- Integer Range—Integer From value to an integer To value.
- IPv6 Address—Value is an IPv6 address.
- IPv6 Prefix—Value is an IPv6 prefix.
- Regular Expression—Value is a Regular Expression in regex syntax. (For common regex usage, see the *"Configuring Administrators" chapter in Cisco Prime Network Registrar 11.2 Administration Guide*.)
- Contains—Value is an IPv6 address or prefix (available for only IPv6 address). The query will list the leases that contain the specified address or prefix.

Step 4 Enter a Value, based on the Type selected. To clear the filter, click **Clear Filter**.

Step 5 Click **Add Element** to add the search element to the Filter Elements list. You can delete the element by expanding the filter display, then clicking the **Delete** icon next to the element.

Step 6 Once you assemble a list of elements, you can search on them, so that the elements are ANDed together for the result. Click **Search**.

Step 7 Check the table of resulting leases from the search, which shows for each an address, state, MAC address, hostname, flags, and expiration date. If necessary, change the page size to see more entries. The leases are ordered by IP address.

CLI Commands

Use **lease list –macaddr** *mac-addr* [**–vpn**=*vpn-name*] to find leases in the DHCPv4 space. Specify the MAC address of the lease. If you omit the VPN designation, you base the search on the current VPN.

For leases in the DHCPv4 space, use the following **lease list** syntax:

```
nrcmd> lease list [-macaddr=mac-addr] [-cm-macaddr=cm-mac-addr]
       [-reservation-lookup-key=key] [-mac | -blob | -string]]
       [-vpn=vpn-name] [-count-only]
```

For leases in the DHCPv4 space, use the following **lease listbrief** syntax:

```
nrcmd> lease listbrief [-macaddr=mac-addr] [-cm-macaddr=cm-mac-addr]
       [-reservation-lookup-key=key] [-mac | -blob | -string]]
       [-vpn=vpn-name] [-count-only]
```

For leases in the DHCPv6 space, use the following **lease6 list** syntax:

```
nrcmd> lease6 list[-duid=client-id]
       [-lookup-key=key] [-blob | -string]]
       [-reservation-lookup-key=key] [-blob | -string]]
       [-macaddr=mac-addr]
       [-cm-macaddr=cm-mac-addr]
       [-vpn=vpn-name] [-count-only]
```

For leases in the DHCPv6 space, use the following **lease6 listbrief** syntax:

```
nrcmd> lease6 listbrief[-duid=client-id]
       [-lookup-key=key] [-blob | -string]]
       [-reservation-lookup-key=key] [-blob | -string]]
       [-macaddr=mac-addr]
       [-cm-macaddr=cm-mac-addr]
       [-vpn=vpn-name] [-count-only]
```

The **–macaddr** and **–cm-macddr** options are to search for leases identified by the CableLabs DOCSIS *vendor-opts* option (DHCPv6 option 17). For example, for these two commands:

```
nrcmd> lease6 listbrief -macaddr=01:02:03:04:05:06
nrcmd> lease6 listbrief -cm-macaddr=01:02:03:04:05:06
```

The –macaddr line lists leases where the option 17 device-id suboption (36) contains the requested MAC address. The –cm-macddr line lists leases where the option 17 cm-mac-address suboption (1026) matches the requested MAC address. (See DHCPv6 Options by Number for details on these suboptions.)

Using Client Reservations

In earlier versions of Cisco Prime Network Registrar versions, the only option for clients to get the lease they want was to create a lease reservation (see Creating Lease Reservations, on page 18). It may not always be easy to create reservations for each client, which may come up to millions of reservations. Also, the process

to update and synchronize the Cisco Prime Network Registrar reservations with databases is very complex. The client reservation feature helps in reducing this complexity.

The current functionality supported by Cisco Prime Network Registrar DHCP server in assigning an IP address to a DHCPv4 client is as follows:

- If a lease based reservation for the client exists and the lease is available, it is used.
- Otherwise, if the client requested an address and it is available, it is used.
- Otherwise, a random address from one of the scopes available to the client is used.

Client reservations feature enables you to supply addresses and delegate prefixes through client entries (either stored directly in Cisco Prime Network Registrar or in LDAP) or through extensions. Also, a client can be located on more than a single scope or prefix and the server will select the address appropriate to the location of the client.

Client-reserved leases are essentially reserved leases. The major difference is that the client for which the lease is reserved is not known to the server in case of client reservations. Client reservations are used when you want to configure leases for many clients or configure many leases for a single client.

Client reservations can be provided to Cisco Prime Network Registrar using one of the following three primary mechanisms:

- Using internal client database—This has some of the same issues as with lease reservations, but may be a better option if Cisco Prime Network Registrar internal client database is already being used for other purposes. The fact that the internal client database has to maintain the client alone and not the reservations makes it more advantageous when compared to lease reservations.
- Using LDAP—Cisco Prime Network Registrar can look up clients in an LDAP repository (external to Cisco Prime Network Registrar) and these clients may specify client reservations.
- Using extensions—Cisco Prime Network Registrar can be set up to communicate with external servers or databases using extensions.

The client entries, maintained either within the Cisco Prime Network Registrar client database or LDAP, can include the addresses and prefixes a client is supposed to use. The attributes to specify the client reservations are:

1. **reserved-addresses**—Specifies the list of addresses reserved for the client. The first available address to match a usable Scope (which must have *restrict-to-reservations* enabled) are assigned to the client.
2. **reserved-ip6addresses**—Specifies the list of addresses reserved for the client. All available addresses to match a usable Prefix (which must have *restrict-to-reservations* enabled) are assigned to the client.
3. **reserved-prefixes**—Specifies the list of prefixes reserved for the client. All available prefixes to match a usable Prefix (which must have *restrict-to-reservations* enabled) are assigned to the client.

Note The above attributes do not indicate a VPN and therefore apply to all VPNs (on which the client may connect). Therefore, if you are using client reservations with VPNs, you either have to assure that the reserved addresses are only valid on the appropriate VPN (as they will apply to all VPNs on which the containing scopes or prefixes exist and has *restrict-to-reservations* enabled) or you need to assure you have unique clients for each VPN.

The attribute *restrict-to-reservations* is added to Scope, Scope template, Prefix, and Prefix template objects to specify the client reservations.

For a client in LDAP, you must set up a mapping between the LDAP attribute name and the corresponding client attribute name.

If the LDAP addresses attribute contained a list of the IPv4 addresses for the client, use **ldap** *servername* **setEntry query-dictionary** *ldap-attribute=cnr-client-attribute* to map it to the *reserved-addresses* attribute. For example:

```
nrcmd> ldap ldap-1 setEntry query-dictionary addresses=reserved-addresses
```

Local Advanced Web UI

To restrict a scope to client reservations, do the following:

1. From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page. See Creating Scopes to create a scope.
2. Click **enabled** for *restrict-to-reservations* attribute in the Miscellaneous Settings group in the List/Add DHCP Scopes page.

   To modify an existing scope to specify client reservations, click the required scope name to open the Edit DHCP Scope page. Click **enabled** for *restrict-to-reservations* attribute in the Miscellaneous Settings group.

   The flag client-reserved shows that a scope is restricted to client reservations.

To restrict a scope template to client reservations, do the following:

1. From the **Design** menu, choose **Scope Templates** under the **DHCPv4** submenu to open the List/Add DHCP Scope Templates page. See Creating and Applying Scope Templates to create a scope template.
2. Click **enabled** for *restrict-to-reservations* attribute in Miscellaneous Settings group in the List/Add DHCP Scope Template page.

   To modify an existing scope template to specify client reservations, click the required scope template name to open the Edit DHCP Scope Template page. Click **enabled** for *restrict-to-reservations* attribute in Miscellaneous Settings group.

To restrict a prefix to client reservations, do the following:

1. From the **Design** menu, choose **Prefixes** under the **DHCPv6** submenu to open the List/Add DHCP v6 Prefixes page.
2. Click the **Add Prefixes** icon in the Prefixes pane, enter the prefix name and address and click the Add IPv6 Prefix.
3. Click the prefix name on the Prefixes pane to open the Edit DHCPv6 Prefix page. Click **enabled** for *restrict-to-reservations* attribute in Non-Parent Settings group.

Note Prefixes which have the *restrict-to-reservations* attribute enabled are not counted in the total of active leases which must be licensed. Any client which receives a client reservation will have that active lease counted, but that will happen only when the lease is actually held by a client.

To restrict a prefix template to client reservations, do the following:

1. To restrict a prefix to client reservations, from the **Design** menu, choose **Prefix Templates** under the **DHCPv6** submenu to open the List/Add DHCP v6 Prefix Templates page.
2. Click the **Add Prefix Templates** icon in the Prefix Templates pane, to open the Add Prefix Template dialog box.
3. Enter the prefix template name and click the **Add Prefix Template** button.
4. Click **enabled** for *restrict-to-reservations* attribute.

   To modify an existing prefix template to specify client reservations, click the prefix template name that you want to restrict to client reservations. Click **enabled** for *restrict-to-reservations* attribute.

Differences Between Client Reservations and Lease Reservations

Client reservations have the following significant differences over lease reservations:

- There is **no** validation to assure that there is only a single client reservation for any address. If there are two different clients that specify the same address or prefix, whichever client request arrives first is granted that lease.
- A client reservation really exists only after the client completes DHCP configuration. Lease reservations are known even if a client transaction never occurs and thus can also be used for clients that do not provide DHCP services at all.

  Cisco Prime Network Registrar supports:

  - Creating a lease reservation for a particular IP address.
  - Configuring the correct cable modem MAC address for the IP address such that Cable Source Verify will work correctly with a Cable Modem Termination System (CMTS).

  This works because the Cisco Prime Network Registrar DHCP server knows about the lease reservation before any DHCP client transaction and will respond correctly to a leasequery request from a CMTS for those addresses. Client reservations are, in contrast, not known to the DHCP server before the arrival of a DHCP client packet at the DHCP server. A leasequery for an IP address which is configured as client-reserved due to some client registration will not (in general) know that the IP address is client reserved.

  Thus, any leasequery to which the DHCP server is supposed to respond with a positive result that includes the proper cable modem MAC address, even when no client has actively requested the lease, will not work with client reservations.

Creating Lease Reservations

To ensure that a client always gets the same lease, you can create a lease reservation. Managing lease reservations is available only to administrators having the dhcp-admin role at the local cluster, or the central-cfg-admin role with the dhcp-management subrole at the regional cluster.

You can query DHCPv4 and DHCPv6 reservations from the server.

Note All lease reservations are counted in the total of active leases that is compared to the number of IP addresses licensed.

DHCPv4 Reservations

When the DHCP edit mode is synchronous, reservation changes are automatically forwarded to the DHCP server, and take immediate effect.

When the edit mode is staged, any change you make to the reservation list on a local cluster modifies the parent scope to indicate that a server reload is required. Any change to the regional reservation list modifies the parent subnet.

Local Basic Web UI

To view lease reservations, from the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page, then click the **Reservations** tab.

To create a reservation on this page, enter the IP address you want to reserve for lease, and enter a lookup key in the Lookup Key field. Click the MAC address (the default) or string or binary radio button, as appropriate for the lookup key entry. Click **Add Reservation**. The lease IP address, Lookup Key and Scope details are displayed in the List/Add DHCP Reservations page.

Local Advanced Web UI

To view the lease reservations for DHCPv4 scopes, from the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page. Proceed as for the Basic web UI.

Advanced mode also provides a mechanism to create reservations independent of scopes. To configure reservations directly for DHCPv4 scopes, do the following:

Step 1 From the **Design** menu, choose **Reservations** under the **DHCPv4** submenu to open the List/Add DHCP Reservations page.

Step 2 Click the **Add DHCP 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, then click **Add Reservation**.

Step 3 Click the MAC address (the default) or string or binary radio button, as appropriate for the lookup key entry. Click **Save**.

Tip You can use a filter to reduce the size of the list that is displayed. To do this, choose a filter type from the **Filter Type** drop-down list. The Filter Value is set as for the selection of the Filter Type. Click **Set Filter**. To set Filter Type as None, click **Clear Filter**. The lease IP address, Lookup Key and Scope details are displayed in the List/Add DHCP Reservations page.

Note Multiple DHCP servers should not distribute IP addresses on the same subnet, unless they are DHCP Failover partners. When using Failover, the client reservations must be identical on each server. If not, a client for whom a lease reservation exists can receive offers of different IP addresses from different servers. The Failover synchronization function helps you assure that the partner configuration is consistent.

CLI Commands

The **reservation** command lets you access the global list of DHCPv4 reservations of Cisco Prime Network Registrar.

Create a new address by using, **reservation** [*vpn-name*/]*address* **create** {*macaddr* | *lookup-key*} [**-mac** | **-blob** | **-string**] [*attribute*=*value* ...]

For example:

```
nrcmd> reservation white/192.168.1.110 create 00:d0:ba:d3:bd:3b
```

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

For example:

```
nrcmd> reservation white/192.168.1.110 delete
```

Get an attribute by using, **reservation** [*vpn-name*/]*address* **get** *attribute*

For example:

```
nrcmd> reservation white/192.168.1.110 get value
```

Set an attribute by using, **reservation** [*vpn-name*/]*address* **set** *attribute*=*value*

For example:

```
nrcmd> reservation white/192.168.1.110 prefix=cm_prefix
```

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

For example:

```
nrcmd> reservation white/192.168.1.110 unset value
```

Show an address by using, **reservation** [*vpn-name*/]*address* **show**

For example:

```
nrcmd> reservation white/192.168.1.110 show
```

Display the reservations by using, **reservation list** [[*vpn-name*/]*address* | **-mac** | **-key**]. This command displays the reservations in *address* order unless -key is specified to change the sort order.

For example:

```
nrcmd> reservation list white/192.168.1.110
```

Display the brief details of the reservations by using, **reservation listbrief** [**-macaddr**=*mac-addr*] [**-lookup-key**=*lookup-key* [**-mac** | **-blob** | **-string**]] [**-vpn**=*vpn-name*] [**-count-only**].

For example:

```
nrcmd> reservation listbrief -lookup-key=d4:6a:a8:d3:e2:ea -mac
```

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 Creating Lease Reservations, on page 18.

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.

Local Advanced Web UI

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 DHCP v6 Reservations page.

Step 2 To create a reservation, click the **Add DHCP v6 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 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 DHCP v6 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**] [*attribute=value* ...]

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 attribute by using, **reservation6** [*vpn-name*/]*address* **get** *attribute*

For example:

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

Set an attribute by using, **reservation6** [*vpn-name*/]*address* **set** *attribute=value*

For example:

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

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

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
```

Display the reservations by using, **reservation6 list** [[*vpn-name*/]*address* | **-key**]. This command displays the reservations in *address* order unless -key is specified to change the sort order.

For example:

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

Display the brief details of the reservations by using, **reservation6 listbrief** [**-lookup-key**=*lookup-key* [**-blob** | **-string**]] [**-vpn**=*vpn-name*] [**-count-only**].

For example:

```
nrcmd> reservation6 listbrief -lookup-key=def -string -vpn=vpn1
```

Setting Advanced Lease and Reservation Properties

Setting advanced lease and reservation properties can include:

- Reserving currently leased IP addresses—See Reserving Currently Leased Addresses, on page 23
- Unreserving leases—See Unreserving Leases, on page 25
- Extending leases to non-MAC addresses—See Extending Reservations to Non-MAC Addresses, on page 25
- Forcing lease availability—See Forcing Lease Availability, on page 27
- Inhibiting lease renewals—See Inhibiting Lease Renewals, on page 28
- Handling leases marked as unavailable—See Handling Leases Marked as Unavailable, on page 31
- Setting timeouts for unavailable leases—See Setting Timeouts for Unavailable Leases, on page 31

Reserving Currently Leased Addresses

You can delete a reservation for one client while reusing it for another one, even though the first client still has the lease.

Local Advanced Web UI

To reserve an existing lease, do the following:

Step 1 From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu, then select the name of the scope to open the Edit DHCP Scope page.

Step 2 Click the **Leases** tab.

Step 3 Click the IP address of the lease.

Step 4 If the IP address is not leased (in available state), enter the lookup key or MAC address for the reservation.

Step 5 Click **Make Reservation**. On the Edit DHCP Scope page, the lease will appear as reserved.

Step 6 Click **Save** to save the scope.

Step 7 To remove the reservation, click **Remove Reservation** on the Edit DHCP Scope page, then modify the scope. The lease no longer appears as reserved.

Example of Reserving an Existing Lease

This CLI command example creates a reservation from an existing lease. It assumes that the dhcp-edit-mode has been set to synchronous to allow the reservations to be added to the server dynamically:

```
nrcmd> reservation 192.168.1.110 create 1,6,00:d0:ba:d3:bd:3b
nrcmd> lease 192.168.1.110 activate
```

Client 1,6,00:d0:ba:d3:bd:3b does a DHCPDISCOVER and gets an offer for 192.168.96.110. The client then does a DHCPREQUEST and gets an ACK message for the same IP address.

As time passes, client 1,6,00:d0:ba:d3:bd:3b does several DHCPREQUESTs that are renewals, which the server acknowledges. Then, at some time before the client lease expiration time, you terminate the reservation:

```
nrcmd> lease 192.168.1.110 deactivate
nrcmd> reservation 192.168.1.110 delete
```

You then add a reservation for a different client for that IP address, even though the address is still leased to the first client:

```
nrcmd> reservation 192.168.1.110 create 1,6,02:01:02:01:02:01
nrcmd> lease 192.168.1.110 activate
```

This action results in an IP address that is leased to one client, but reserved for another. If the new client (1,6,02:01:02:01:02:01) does a DHCPDISCOVER before the original client (1,6,00:d0:ba:d3:bd:3b) does, the new client does not get 192.168.96.110, but gets a random IP address from the dynamic pool.

When the original client (1,6,00:d0:ba:d3:bd:3b) sends its next DHCPREQUEST/RENEW for the lease on 192.168.96.110, it gets a NAK message. Generally, upon receipt of the not-acknowledged message, the client immediately sends a DHCPDISCOVER. On receiving that DHCPDISCOVER, the server cancels the remaining lease time for 192.168.96.110.

The server then gives client 1,6,00:d0:ba:d3:bd:3b whatever lease is appropriate for it—some reservation other than 192.168.96.110, some dynamic lease (if one is available), or nothing (if no dynamic leases are available). When the new client (1,6,02:01:02:01:02:01) tries to renew the random IP address it received, the server sends it a NAK, because it wants to give it the reserved address. When the new client then does a DHCPDISCOVER, it gets the 192.168.96.110 reserved address.

You could also force availability of a lease (see Forcing Lease Availability, on page 27). However, doing so does not stop the original client (1,6,00:d0:ba:d3:bd:3b) from using 192.168.96.110. Also, it does not prevent the new client (1,6,02:01:02:01:02:01) from getting 192.168.96.110. In other words, this means that making a reservation for a client is independent of the lease state (and actual lease client) of the IP address for which the reservation is made.

Thus, making a reservation for one client does not cause another client to lose that lease right away, although that client receives a NAK response the next time it contacts the DHCP server (which could be seconds or days). Additionally, the client that reserved the IP address does not get that address if some other client already has it. Instead, it gets another IP address until the:

- IP address it is supposed to receive is free.

- Client sends a DHCPREQUEST as a renewal and receives a NAK response.
- Client sends a DHCPDISCOVER.

Unreserving Leases

You can remove lease reservations at any time. However, if the lease is still active, the client continues to use the lease until it expires. If you try to reserve the lease for a different client, you will get a warning.

Once you delete the last reservation from regional, it is not possible to select the reservation and push the change to the local cluster. You must push the parent subnet, which will then synchronize the reservation list and thus delete the local copy of the reservation.

There is no push function for DHCPv6 reservations in regional. You always need to push the parent prefix to resynchronize the reservations. This is the preferred method when synchronizing regional delete actions.

Local Advanced Web UI

To unreserve a lease, from the **Design** menu, choose **Reservations** under the **DHCPv4** submenu to open the List/Add DHCP Reservations page, then click the **Delete Reservation** icon (on the leftpane) after selecting the reservation you want to remove. This removes the reservation immediately, with no confirmation.

CLI Commands

To unreserve a lease, use **reservation** [*vpn/*]*ipaddr* **delete** or **scope** *name* **removeReservation** {*ipaddr* | *macaddr* | *lookupkey*} [**–mac** | **–blob** | **–string**]. However:

- Ensure that the reservation is gone from the nrcmd internal database.
- If you use failover on the scope containing the reservation:

1. Use **reservation** [*vpn/*]*ipaddr* **delete**, or **scope** *name* **removeReservation**, on both servers.
2. On the backup server, if you are in staged dhcp edit mode, use **lease** [*vpn/*]*ipaddr* **delete-reservation**.
3. Use the same command on the main server.

Save the result of this operation to preserve it across server reloads, because issuing **lease** *ipaddr* **delete-reservation** alone affects only the server internal memory.

Extending Reservations to Non-MAC Addresses

You might need to create lease reservations based on something other than the MAC address from the incoming client packet. Often, DHCP client devices attached to a switch port need to get the same IP address, regardless of the MAC address. This approach helps when you replace factory floor devices with identical devices (with different MAC addresses), but want to maintain the same IP address.

Overriding Client IDs

You can set an expression in a client-class *override-client-id* attribute that extracts the MAC address and port of a switch from the relay-agent-info option (82) and creates a client identity from it. Regardless of the client-id in the incoming packet, the identity that allocates an IP address is the same for any device coming in through the same switch port. The expression you use for the attribute depends on the option 82 format. The DHCP server calculates the expression when it assigns the packet to the client-class. The *override-client-id* value becomes the identity of the client from that point onward.

Note When using *[v6-]override-client-id* expressions, leasequery by client-id requests may need to specify the *override-client-id* attribute to correctly retrieve the information on the lease(s) for the client.

However, when you enable the *use-client-id-for-reservations* attribute in a policy, the server turns the client-id of that request into a string of the form *nn*:*nn*:*nn* ... *nn*:*nn*, and uses that string to look up the reservation.

The *add-to-environment-dictionary* attribute for a client or client-class also serves to send attribute values to the DHCP extension environment dictionary (see Using Extension Points), specified as name-value pairs. You can configure an *add-to-environment-dictionary* attribute on either a client or a client-class. If you choose to configure this attribute on both a client and client-class, you should ensure that the name-value pairs that you configure on the client have different names than the name-value pairs that you configure on the client-class, because they are all going to be put into the same environment dictionary (which can have only one value for a particular name). Generally, it is best to configure this attribute on a client or a client-class only, but not on both.

Local Advanced Web UI

You can find the *override-client-id* attribute on the Edit DHCP Client Class page (from the **Design** menu, choose **Client Classes** under the **DHCP Settings** submenu, then the name of the client-class).

You also need to configure a client-class lookup ID for the DHCP server, to put every packet into a particular client-class where you configure the *override-client-id* expression. From the **Operate** menu, choose **Manage Servers** under the **Servers** submenu, then click **DHCP** to open the Edit Local DHCP Server page. In the Client Class attributes, enter a *client-class-lookup-id* expression.

To use the client ID for the reservation, configure the policy to enable the *use-client-id-for-reservations* attribute on the Add DHCP Policy page (from the **Design** menu, choose **Policies** under the **DHCP Settings** submenu, then click **Add Policies**) or Edit DHCP Policy page (from the **Design** menu, choose **Policies** under the **DHCP Settings** submenu, then the name of the policy).

CLI Commands

The syntax for setting the *override-client-id* attribute is **client-class** *name* **set override-client-id="***expression***"**. The syntax for setting the *client-class-lookup-id* attribute is **dhcp set client-class-lookup-id="***expression***"**. The syntax for setting the *use-client-id-for-reservations* attribute is **policy name enable use-client-id-for-reservations**.

Reservation Override Example

The following example shows how to override a client ID for a reservation:

Step 1 Create a scope for the reservation:

a) Enter a subnet address.
b) If you want dynamic reservations, add an IP address range.

Step 2 Add the reservation for the scope:

a) Include a value for the lookup key.
b) Specify the lookup key type as binary.

Step 3 Create a policy for the purpose, enabling the *use-client-id-reservations* attribute.

Step 4 Create a client-class for the purpose:

a) Specify the policy created in the previous step.

b) Include an expression for the *override-client-id* attribute that returns a blob value with the client ID you want, based on the contents of the packet.

Step 5 Get a lease for a client with the MAC address. This client will then get the override ID.

Reconfiguring IPv6 Leases

For DHCPv6 leases, you 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.

For more information on enabling the DHCPv6 policy reconfiguration, see Configuring DHCPv6 Policies.

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 [vpn-name/]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 **lease6 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.

Forcing Lease Availability

You can force a current lease to become available. You should request that the user release the lease, or do so yourself, before forcing its availability. Forcing lease availability does not require a server reload.

Note After a lease is forced to be available, the client continues to use it until the client contacts the DHCP server.

Local Advanced Web UI

To force lease availability, do the following:

Step 1 From the **Design** menu, choose **Scopes** under the **DHCPv4** submenu to open the List/Add DHCP Scopes page.

Step 2 Click the **Lease** tab for the scope that has leases.

Step 3 Click the IP address of the lease on the Edit DHCP Scope page.

Step 4 Click **Force Available** on the Edit DHCP Scope page. The lease will now show an empty value in the Flags column.

CLI Commands

To force lease availability, use **lease** [*vpn/*]*ipaddr* **force-available**. Use **scope** *name* **clearUnavailable** to force all "**Unavailable**" leases in the scope to change to "**Available**" states.

Inhibiting Lease Renewals

Normally, the Cisco Prime Network Registrar DHCP server retains the association between a client and its leased IP address. The DHCP protocol explicitly recommends this association and it is a usually desirable feature. However, for some customers, such as ISPs, clients with long-lived lease associations may be undesirable, because these clients should change their IP addresses periodically. Cisco Prime Network Registrar includes a feature that allows customers to force lease associations to change when DHCP clients attempt to renew their leases or reboot.

A server can never force a client to change its lease, but can compel the client to do so based on a DHCPRENEW or DHCPDISCOVER request. Cisco Prime Network Registrar offers configuration options to allow customers to choose which interactions to use to force a client to change its IP address:

- **Inhibiting all lease renewals**—While a client is using a leased address, it periodically tries to extend its lease. At each renewal attempt, the server can reject the lease, forcing the client to stop using the IP address. The client might have active connections that are terminated when the lease terminates, so that renewal inhibition at this point in the DHCP interaction is likely to be user-visible.
- **Inhibiting renewals at reboot**—When a DHCP client reboots, it might have recorded a valid lease binding that did not expire, or it might not have a valid lease. If it does not have a lease, you can prevent the server from granting the last held lease. If the client has a valid lease, the server rejects it, forcing the client to obtain a new one. In either case, no active connections can use the leased address, so that the inhibition does not have a visible impact.
- **Effect on reservations**—Reservations take precedence over renewal inhibition. If a client has a reservation, it can continue to use the reserved IP address, whether or not renewal inhibition is configured.
- **Effect on client-classes**—Client-class testing takes place after renewal inhibition testing. If a client is forced to change IP addresses by renewal inhibition, then client-class processing might influence which address the server offers to the client.

You can enable or disable lease renewal inhibition for a policy, which you can set system wide, for a scope or on a client-by-client basis. The *inhibit-all-renews* attribute causes the server to reject all renewal requests, forcing the client to obtain a new IP address any time it contacts the DHCP server. The *inhibit-renews-at-reboot* attribute permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot. It applies only to DISCOVER and INIT-REBOOT operations. DISCOVER is included because few DHCP clients use INIT-REBOOT when they reboot (most clients just do a DISCOVER).

Renewals are not inhibited under the following conditions:

- When using failover and the time elapsed since the *start-time-of-state* is less than the MCLT. The default MCLT is 60m.
- When using failover, and the failover state is not NORMAL or PARTNER-DOWN.
- When the lease is AVAILABLE and the time elapsed since the *client-creation-time* is less than the *renewal-inhibition-max-time*. The default value of *renewal-inhibition-max-time* is 60s.
- When the lease is OFFERED or LEASED, and the request is DISCOVER or REQUEST-SELECTING, and the time elapsed since the *start-time-of-state* is less than the *renewal-inhibition-max-time*. The default value of *renewal-inhibition-max-time* is 60s.

The DHCP server needs to distinguish between a client message that it should reject (such as a renewal request) and one that represents a retransmission. When the server processes a message, it records the time the packet arrived. It also records the time at which it made a lease binding to a client, and the last time it processed a message from the client about that binding. It then compares the packet arrival time with the lease binding time (the *start-time-of-state*) and processes packets from the client within a certain time interval from the start time of the binding. By default, this time interval is one minute.

Local Advanced Web UI

To inhibit lease renewals, create a policy on the Edit DHCP Policy page (from the **Design** menu, choose **Policies** under the **DHCP Settings** submenu, then the name of the policy), then enable the *inhibit-all-renews* or *inhibit-renews-at-reboot* attribute. (Both attributes are preset to disabled). Then, modify the policy and click **Save** to save the changes.

Moving Leases Between Servers

There may be a need to move leases to a new DHCP server such as, the configuration of the server grows sufficiently large to exceed the recommended limits. There are different ways to accomplish this task depending on whether the leases are being moved to a new server or an existing server. Either of these techniques requires special considerations and careful execution. A new server is often the simplest way to accomplish by moving the entire configuration and the state database. To move the leases to another server, the leaseadmin utility is used. This utility allows you to export all or a selected set of leases and also to import this exported lease set.

Caution The leaseadmin utility must only be used on a local cluster (exporting or importing) and the DHCP server must be stopped before running the leaseadmin utility.

The leaseadmin utility was added to Cisco Prime Network Registrar to allow leases to be moved from one server to another. You must run this utility on the same machine as the DHCP server and you must have superuser/root privileges to read and modify the database file. This utility requires direct access to the lease state database; however, stopping the DHCP server is not sufficient as the stopped server still holds the lease state database open. If the utility is run when the database is still in use, the leaseadmin utility will report the error "Failed to obtain exclusive access to lease state database". The default location is:

/opt/nwreg2/local/usrbin

From the command prompt, change to the above location and run the utility using the syntax:

./leaseadmin <*options*>

The table below describes the qualifying options for the **leaseadmin** utility.

Table 3: leaseadmin Command Options

Option	Description
To export lease(s)	
–e *filename*	Exports to a file.
–x	Sends raw output format (required to import).
–t {**current** | **history** | **detail** | **all** | **v6leases** | **v6history**}	Specifies the record types to be exported. Valid values are: **current**, **history**, **detail**, **all**, **v6leases**, and **v6history**
–s *subnet* | *prefix*	Restrict the lease records to be exported to a subnet or a prefix.
To import lease(s)	
–i *filename*	Import from a file. When used with **-n** option, specifies the VPN.
–o	When used with the –i (import) option, overwrites the existing data.
–c	Compress records.
To delete lease(s) or the server DHCP Unique Identifier (DUID)	
–d *address* | *subnet* | *prefix*	Specifies the address, subnet, or prefix to be deleted.
–d server-duid	Specifies that the server DUID information is to be deleted from the database. **Note** When you specify the server-duid, the auto-generated DHCPv6 server DUID is deleted, if it exists. While not recommended (see Things to Avoid When Troubleshooting Issues Related to Failover), if lease databases are ever copied to another local cluster, it is critical that any server-duid that may be present in the copied database is deleted from it using this operation. If the server-duid is deleted, when the DHCP server starts, it will generate a new server-duid. This will cause all DHCPv6 Renew requests specifying the older server-duid as the server-id option to be dropped until the client starts sending DHCPv6 Rebind requests. For 10.x and later only, once deleted, the server uses the local cluster's UUID and this is not stored in the lease database as it is available from the cluster's configuration.
General Options	
–n *vpn*	When used with -e (export), -i (import), or -d (delete) option, specifies the VPN. To include all VPNs specify "all".
–h *path*	Overrides the default path to the database.
–v	Displays the database version.

Option	Description
–z *{letters}=level*	Sets the debug output levels.

Handling Leases Marked as Unavailable

One of the aspects of effective lease maintenance is determining the number of unavailable leases in a scope. This number is sometimes higher than expected. Each unavailable lease is probably an indication of a serious problem. Possible causes for an unavailable lease are:

- **The DHCP server is configured for a ping before an offer, and the ICMP echo message is returned successfully**—A currently active client is using that IP address, causing the DHCP server to mark it as *unavailable*. To prevent the server from doing so, disable pinging an address before offering it to a client. See Pinging Hosts Before Offering Addresses, on page 10.
- **The server receives a DHCPDECLINE message from a client to which it leased what it considered to be a good IP address**—The client does an address resolution (ARP) request for the IP address on its local LAN segment, and another client responds to it. The client then returns the address to the server with a DHCPDECLINE packet and sends another DHCPDISCOVER packet to get a new address. The server marks as *unavailable* the address that the client returns. To prevent the server from reacting to DHCPDECLINE messages, you can set a scope attribute, *ignore-declines*.
- **The server receives "other server" requests from the client**—Because all DHCPREQUEST messages that follow DHCPOFFER messages are broadcast, the server can see messages directed to other DHCP servers. A server knows that a message is directed to it by the value of the *server-id* option in the packet. If the Cisco Prime Network Registrar server recognizes a message directed at another server, in that its own IP address does not appear in the *server-id* option, but the address leased in the message is one that the server controls, it believes that two servers must be trying to manage the address simultaneously. It then marks the local address as *unavailable*. This behavior does not apply in a DHCP failover configuration. Either the two servers are configured with some or all of the same IP addresses, or (in rare cases) the DHCP client placed a wrong *server-id* option value in the packet.

  If you have reason to believe that the client is sending bad *server-id* options (rather than packets actually directed to other servers), Cisco Prime Network Registrar has a server attribute you can enable that turns this behavior off, the *ignore-requests-for-other-servers* attribute.

- **Inconsistent lease data**—Extremely rare and occurring only during server startup when, while configuring a lease, the server reads the lease data from disk during a refresh of the internal cache. The lease state appears as *leased*, but there is incomplete data to construct a client for that lease, such as that the lease might not yet have a *client-id* option value. The server considers the data to be inconsistent and marks the IP address as *unavailable*. Forcing the lease to be available (such as by using the **lease** *ipaddr* **force-available** command in the CLI) should clear up this problem.

Setting Timeouts for Unavailable Leases

During the times when leases become unavailable, as described in Handling Leases Marked as Unavailable, on page 31, all unavailable leases remain in that state for a configured time only, after which time they again become available. A policy attribute, *unavailable-timeout*, controls this time. The *system_default_policy* policy sets this value to one day by default.

To handle upgrades from previous releases of Cisco Prime Network Registrar that do not have this timeout feature, a special upgrade timeout attribute, *upgrade-unavailable-timeout* (which is preset to one day) is included at the server level. The *upgrade-unavailable-timeout* value is the timeout given to leases set to unavailable before the Cisco Prime Network Registrar upgrade. This setting affects the running server only and does not rewrite the database. If the server stays up for one day without reloading, all the unavailable leases that were present at the last reload will time out. If the server reloads in less than a day, the entire process restarts with the next reload. Note that this process occurs only for leases that were set unavailable before the upgrade. Leases that become unavailable after the upgrade receive the *unavailable-timeout* value from the policy, as previously described.

Querying Leases

Cisco Prime Network Registrar can work together with Cisco routers to provide enhanced provisioning capabilities. This function is described in the DHCP Leasequery specification (RFC 4388), with which Cisco Prime Network Registrar conforms. Part of the implementation of the Cisco uBR access concentrator relay agent is to capture and glean information from DHCP lease requests and responses. It uses this information to:

- Associate subscriber cable modem and client MAC addresses with server-assigned IP addresses.
- Verify source IP addresses in upstream datagrams.
- Encrypt unicast downstream traffic through the DOCSIS Baseline Privacy protocol.
- Avoid broadcasting downstream Address Resolution Protocol (ARP) requests, which can burden the the uBR as well as the subscriber hosts, and which malicious clients can compromise.

The uBR device does not capture all DHCP state information through gleaning. The uBR device cannot glean from unicast messages (particularly renewals and releases) because capturing them requires special processing that would degrade its forwarding performance. Also, this data does not persist across uBR reboots or replacements. Therefore, the only reliable source of DHCP state information for the uBR device is the DHCP server itself.

For this reason the DHCP server supports the DHCPLEASEQUERY message, which is similar to a DHCPINFORM message. Access concentrators and relay agents can thereby obtain client location data directly from the DHCP server, for DHCPv4 and DHCPv6 addresses.

Leasequery Implementations

Cisco Prime Network Registrar provides three Leasequery implementations:

- DHCPv4 Cisco-proprietary for pre-RFC 4388—See Pre-RFC Leasequery for DHCPv4, on page 33
- DHCPv4 compliant with RFC 4388—See RFC 4388 Leasequery for DHCPv4, on page 34
- DHCPv6—See Leasequery for DHCPv6, on page 34

The Cisco-proprietary and the more recent RFC-compliant implementations for DHCPv4 differ in only minor ways and will coexist. The DHCP server accepts Leasequery requests at the same port and returns the specified data for both implementations. The DHCPv6 implementation conforms with RFC 5007 and RFC 5460.

The DHCP server can include lease reservation data in Leasequery responses for DHCPv4 and DHCPv6. Cisco Prime Network Registrar returns a default lease time of one year (31536000 seconds) for reserved DHCPv4 and lifetime of the leases for DHCPv6 leases in a response. If the IP address is actually leased, Cisco Prime Network Registrar returns its remaining lease time.

Leasequery is preset to be enabled for all the implementations. To disable it, disable an Expert mode attribute, *leasequery*.

Pre-RFC Leasequery for DHCPv4

Leasequery messages usually contain request fields and options. To illustrate, suppose that after a relay agent reboot or replacement, the relay agent receives a request to forward a datagram downstream to the public broadband access network. Because the relay agent no longer has the downstream location data, it sends a LEASEQUERY message to the DHCP server that includes the gateway IP address (*giaddr*) of the relay agent and the MAC address or *dhcp-client-identifier* (option 61) of the target client. If the DHCP server finds the client, it returns the client IP address in the client address (*ciaddr*) field in the response to the leasequery. If the server cannot find the client address, it returns a DHCPNACK.

In the pre-RFC implementation for DHCPv4, the requestor can query by IP address, client ID option (61), or MAC address, and receives from the server a DHCPACK (with the returned data) or a DHCPNACK message, or the server drops the packet. If the request includes multiple query types, the DHCP server responds to the first one it can find. The *giaddr* value from the requestor is independent of the *ciaddr* searched and is simply the return IP address for any responses from the server. The three possible query types are:

- **IP address (***ciaddr***)**—The request packet includes an IP address in the *ciaddr* field. The DHCP server returns data for the most recent client to use that address. A packet that includes a *ciaddr* value must be a request by IP address, despite the values in the MAC address fields (*htype*, *hlen*, and *chaddr*) or *dhcp-client-identifier* option. Querying by IP address is the most efficient method and the one most widely used, in that the other two methods can put more load on the DHCP server.
- **dhcp-client-identifier option (61)**—The request packet includes a *dhcp-client-identifier* option value. The DHCP server returns a DHCPACK packet containing the IP address data for the most recently accessed client. If the request omits a MAC address, the server returns all IP addresses and their data for the requested client ID in the *cisco-leased-ip* (also called the *associated-ip*) option. If the request includes the MAC address, the server matches the *dhcp-client-identifier* and MAC address with the client data for the IP address and returns that data in the *ciaddr* field or *cisco-leased-ip* (also called the *associated-ip*) option.
- **MAC address**—The request packet includes a MAC address in the hardware type (*htype*), address length (*hlen*), and client hardware address (*chaddr*) fields, and a blank *ciaddr* field. The server returns all the IP addresses and most recent lease data for the MAC address in the *cisco-leased-ip* (also called the *associated-ip*) option of the reply packet.

The DHCPLEASEQUERY message number in the *dhcp-message-type* option (53) for the pre-RFC implementation is 13. A server that does not support this type of message is likely to drop the packet. The DHCPACK message reply always contains the physical address of the lease owner in the *htype*, *hlen*, and *chaddr* fields. If the request contains the *ciaddr*, the data returned is always based on the *ciaddr* and never the client ID or MAC address.

The requestor can include the *parameter-request-list* option (55) to request specific options about an address. The reply often contains the *dhcp-lease-time* option (51) and the original content of the *relay-agent-info* option (82) that the client sent. If the server does not detect a valid lease for a client, it does not return option 51, and the requestor needs to determine if there is a valid lease.

A DHCPACK from the server can also contain the following Leasequery options:

- ***cisco-leased-ip* (161)**—Data on all the IP addresses associated with the client; also known as (and later renamed) the *associated-ip* option.
- ***cisco-client-requested-host-name* (162)**—Hostname that the client requested in the *host-name* option (12) or *client-fqdn* option (81). The requested hostname was dropped in the RFC 4388 implementation.

- ***cisco-client-last-transaction-time* (163)**—Most recent time duration that a DHCP server contacted the client.

RFC 4388 Leasequery for DHCPv4

Leasequery became an official RFC 4388 for DHCPv4 in February 2006. Cisco Prime Network Registrar provides the RFC 4388 implementation alongside the pre-RFC one (see Pre-RFC Leasequery for DHCPv4, on page 33) and there are no conflicts between them. However, the RFC 4388 implementation includes a few notable changes:

- The DHCPLEASEQUERY message type contained in the *dhcp-message-type* option (53) changed its message ID to 10 (the ID 13 was given to the DHCPLEASEACTIVE message), and the reply messages were changed from just DHCPACK and DHCPNACK to be more specific:
  - DHCPLEASEQUERY (10) for queries
  - DHCPLEASEUNASSIGNED (11) for replies of unassigned addresses
  - DHCPLEASEUNKNOWN (12) for replies of unknown addresses
  - DHCPLEASEACTIVE (13) for replies of active addresses
- The reply option names and IDs changed, and the *cisco-client-requested-host-name* option was dropped so that there are only two reply options:
  - ***client-last-transaction-time* (91)**—Most recent time duration that a DHCP server contacted the client.
  - ***associated-ip* (92)**—Data on all the IP addresses associated with the client.
- If querying by client ID or MAC address, the request can contain only the *dhcp-client-identifier* option (61) or MAC address; if the packet contains both, the server drops it.

Leasequery for DHCPv6

Cisco Prime Network Registrar supports both the RFC 5007 (UDP) and RFC 5460 (TCP, Bulk) DHCPv6 leasequery capabilities.

Note To use the RFC 5460 (TCP, Bulk) leasequery support, you must create a DHCP Listener for IPv6 (see DHCP Listener Configuration, on page 52).

The message types for DHCPv6 Leasequery are:

- LEASEQUERY (14)
- LEASEQUERY_REPLY (15)
- LEASEQUERY_DATA (17)
- LEASEQUERY_DONE (16)
- ACTIVELEASEQUERY(240)

A query can be by:

- QUERY_BY_ADDRESS (1)
- QUERY_BY_CLIENTID (2)
- QUERY_BY_RELAY_ID(3)
- QUERY_BY_LINK_ADDRESS(4)

- QUERY_BY_REMOTE_ID(5)

A DHCPv6 LEASEQUERY_REPLY message can contain one or more of the following options:

- ***lq-query* (44)**—Query being performed. The option, used in a request only, includes the query type, link-address (or 0::0), and options to provide data needed for the query.
- ***client-data* (45)**—Encapsulates the data for a single client on a single link. The client data can include any number of these or other requested options.
- ***clt-time* (46)**—Client last transaction time encapsulated in a *client-data* option (45); identifies how long ago (in seconds) the server last communicated with the client.
- ***lq-relay-data* (47)**—Relay agent data used when the client last communicated with the server. Fields are the peer-address and the relay-message. This option can include further options.
- ***lq-client-link* (48)**—Links on which the client has any bindings. Used in reply to a client query when the link-address is omitted and the client is found to be on more than one link.
- ***option_lq_base_time***—Specifies the current absolute time on DHCPv6 server at the time of sending binding information.

A DHCPv6 LEASEQUERY_REQUEST message can contain one or more of the following options:

- ***option_lq_start_time***—Bindings updated since the specified time. This option, used for the list of binding updates happened during the offline period.
- ***option_lq_end_time***—Bindings updated during the specified time period.

DHCPv6 uses the Option Request option (*oro*) to request a list of options in the Leasequery reply.

Note Leasequery by client-id requests may need to specify the *override-client-id* attribute when using [*v6-override-client-id*] expressions to correctly retrieve the information on the lease(s) for the client.

Leasequery Statistics

Lease queries provide statistics attributes, in the web UI, on the DHCP Server Statistics page (see the *"Displaying Statistics" section in Cisco Prime Network Registrar 11.2 Administration Guide*), and in the CLI by using **dhcp getStats**. The Leasequery statistics are:

- **lease-queries**—Number of RFC 4388 message ID 10 (or pre-RFC message ID 13) DHCPv4 Leasequery packets received in the given time interval.
- **lease-queries-active**—Number of RFC 4388 DHCPLEASEACTIVE packets.
- **lease-queries-unassigned**—Number of RFC 4388 DHCPLEASEUNASSIGNED packets.
- **lease-queries-unknown**—Number of RFC 4388 DHCPLEASEUNKNOWN packets.
- **leasequeries**—Number of DHCPv6 Leasequery packets received.
- **leasequery-replies**—Number of responses to DHCPv6 Leasequery packets that might or might not be successful.
- **tcp-current-connections**—Number of currently open TCP connections to the DHCP server for DHCPv6 Active and Bulk Leasequery.
- **tcp-total-connections**—Number of TCP connections that were opened to the DHCP server for DHCPv6 Active and Bulk Leasequery in this time interval.

- **bulk-leasequeries**—Number of LEASEQUERY packets received over all TCP connections in this time**bulk-leasequeries** interval.
- **bulk-leasequery-replies**—Number of LEASEQUERY-REPLY packets sent over all TCP connections in this time interval.
- **bulk-leasequery-data**—Number of LEASEQUERY-DATA packets sent over all TCP connections in this time interval.
- **bulk-leasequery-done**—Number of LEASEQUERY-DONE packets sent over all TCP connections in this time interval.
- **tcp-lq-status-unspec-fail**—Number of LEASEQUERY-REPLY packets with a status code of UnspecFail(1) sent over TCP in this time interval.
- **tcp-lq-status-unknown-query**—Number of LEASEQUERY-REPLY packets with a status code of UnknownQueryType(7) sent over TCP in this time interval.
- **tcp-lq-status-malformed-query**—Number of LEASEQUERY-REPLY packets with a status code of MalformedQuery(8) sent over TCP in this time interval.
- **tcp-lq-status-not-configured**—Number of LEASEQUERY-REPLY packets with a status code of NotConfigured(9) sent over TCP in this time interval.
- **tcp-lq-status-not-allowed**—Number of LEASEQUERY-REPLY packets with a status code of NotAllowed(10) sent over TCP in this time interval.
- **tcp-lq-status-query-terminated**—Number of LEASEQUERY-REPLY/LEASEQUERY-DONE packets with a status code of QueryTerminated(11) sent over TCP in this time interval.
- **tcp-connections-dropped**—Number of TCP requests that were terminated in this time interval because the TCP connection was closed (or reset) by the DHCPv6 requester. This excludes normal connection closes or server reloads.
- **active-leasequeries**—Number of ACTIVELEASEQUERY packets received over all TCP connections in this time interval.
- **active-leasequery-replies**—Number of LEASEQUERY-REPLY packets sent over all TCP connections in this time interval for active leasequery.
- **active-leasequery-data**—Number of LEASEQUERY-DATA packets sent over all TCP connections in this time interval for active leasequery.
- **active-leasequery-done**—Number of LEASEQUERY-DONE packets sent over all TCP connections in this time interval for active leasequery.
- **tcp-lq-status-data-missing**—Number of LEASEQUERY-REPLY packets with a status code of DataMissing(240) sent over TCP in this time interval.
- **tcp-lq-status-catch-up-complete**—Number of LEASEQUERY-DATA packets with a status code of CatchUpComplete(241) sent over TCP in this time interval.

Leasequery Example

The example below shows a packet trace of a DHCPv6 UDP query by client ID without a link-address, but with addresses on more than one link. The first part of the output shows the query message and the second part shows the reply data. The *lq-query* option identifies the type of query. Note the list of requested options via the Option Request option (*oro*) in the request, and the two addresses returned in the *lq-client-links* option in the reply.

Example: Packet Trace of Sample UDP Lease Query

```
+- Start of LEASEQUERY (14) message (113 bytes)
| transaction-id 22
| lq-query (44) option (37 bytes)
| (query-type 2, link-address ::)
```

```
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| oro (6) option (2 bytes)
| 47
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| vendor-class (16) option (14 bytes)
| (enterprise-id 1760,
| ((00:08:41:49:43:20:45:63:68:6f)))
| vendor-class (16) option (14 bytes)
| (enterprise-id 1760,
| ((00:08:41:49:43:20:45:63:68:6f)))
+- End of LEASEQUERY message
+- Start of LEASEQUERY-REPLY (15) message (72 bytes)
| transaction-id 22
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| lq-client-links (48) option (32 bytes)
| 2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e,2001:4f8:ff00:0:e400:f92:1bfd:60fa
+- End of LEASEQUERY-REPLY message
```

The example below shows a packet trace of a DHCPv6 TCP query by client ID. The first part of the output shows the request message, the second part shows the response message with the binding data of the first client, and the last part will show that the query has ended successfully. The third part will follow the second part if there are more than a single client to be returned.

Note The LEASEQUERY-DONE message will not be present in a packet if the LEASEQUERY-REPLY message does not have any binding data.

Example: Packet Trace of Sample TCP Lease Query

```
+- Start of LEASEQUERY (14) message (59 bytes)
| transaction-id 2
| lq-query (44) option (37 bytes)
| (query-type 2, link-address ::)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| oro (6) option (2 bytes)
| 47
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
+- End of LEASEQUERY message

+- Start of LEASEQUERY-REPLY (15) message (162 bytes)
| transaction-id 2
| server-identifier (2) option (14 bytes)
| 00:01:00:01:13:06:6a:67:00:23:7d:53:e5:e3
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:03:05:07:09:11
| client-data (45) option (122 bytes)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| clt-time (46) option (4 bytes)
| 5m54s
```

```
| iaaddr (5) option (24 bytes)
| (address 2001:4f8:ffff:0:8125:ef1b:bdcb:4b4e,
| preferred-lifetime 6d23h54m6s,
| valid-lifetime 1w6d23h54m6s)
| lq-relay-data (47) option (68 bytes)
| peer-address fcc0:a803::214:4fff:fec1:226a
| +- Start of RELAY-FORW (12) message (52 bytes)
| | hop-count 0,
| | link-address 2001:4f8:ffff::,
| | peer-address fe80::302:3ff:fe04:506
| | vendor-class (16) option (14 bytes)
| | (enterprise-id 1760,
| | ((00:08:41:49:43:20:45:63:68:6f)))
| +- End of RELAY-FORW message
+- End of LEASEQUERY-REPLY message
+- Start of LEASEQUERY-DATA (17) message (130 bytes)
| transaction-id 2
| client-data (45) option (122 bytes)
| client-identifier (1) option (10 bytes)
| 00:03:00:01:01:02:03:04:05:06
| clt-time (46) option (4 bytes)
| 5m33s
| iaaddr (5) option (24 bytes)
| (address 2001:4f8:ff00:0:e400:f92:1bfd:60fa,
| preferred-lifetime 6d23h54m27s,
| valid-lifetime 1w6d23h54m27s)
| lq-relay-data (47) option (68 bytes)
| peer-address fcc0:a803::214:4fff:fec1:226a
| +- Start of RELAY-FORW (12) message (52 bytes)
| | hop-count 0,
| | link-address 2001:4f8:ff00::,
| | peer-address fe80::302:3ff:fe04:506
| | vendor-class (16) option (14 bytes)
| | (enterprise-id 1760,
| | ((00:08:41:49:43:20:45:63:68:6f)))
| +- End of RELAY-FORW message
+- End of LEASEQUERY-DATA message

+- Start of LEASEQUERY-DONE (16) message (4 bytes)
| transaction-id 2
+- End of LEASEQUERY-DONE message
```

Difference between TCP bulk leasequery and UDP leasequery

The following are the differences between TCP bulk leasequery and UDP leasequery:

- UDP leasequery supports Query by IPv6 Address and Query by Client Identifier. However, TCP Bulk Leasequery supports all the five query types; that is, Query by IPv6 Address, Query by Client Identifier, Query by Relay Identifier, Query by Link Address, and Query by Remote ID.
- In UDP Leasequery, if the server finds bindings for the relay agent on multiple links, then DHCP server will send an option OPTION_CLIENT_LINK in the reply message. The relay agent will need to resend LEASEQUERY messages using each of the returned link-addresses to obtain the all client's bindings. Whereas in TCP Bulk Leasequery, the server returns multiple bindings of a client on different links; however OPTION_CLIENT_LINK is not supported in Bulk Leasequery reply.

Running Address and Lease Reports

You can run these reports on IP addresses and leases:

- **Address Usage**—See Running Address Usage Reports, on page 39
- **Lease History**—See Running IP Lease Histories, on page 39
- **Current Utilization**—See Running Lease Utilization Reports, on page 45
- **Lease Notification**—See Receiving Lease Notification, on page 45

Running Address Usage Reports

The address usage reports show the IP addresses that are assigned leases.

Local Advanced Web UI

To view the leases for IP addresses, on the Edit DHCP Scope page (from the **Design** menu, choose **Scopes** under the **DHCPv4** submenu), click the **Leases** tab to open the List DHCP Leases for the scope. To manage a specific lease, click its IP address on the page.

CLI Commands

To view the IP address usage for specified servers, use **report**.

Tip If you are not already using **lease-notification** in an automated way, try **lease-notification available=100%** for a concise scope-by-scope summary of the state of the servers.

Running IP Lease Histories

You can extract IP lease history data from a special database so that you can determine past allocation information for a given IP address. You can get a historical view of when a client was issued a lease, for how long, when the client or server released the lease before it expired, and if and when the server renewed the lease and for how long.

Cisco Prime Network Registrar provides a client to control querying IP history data. Through this client, you can:

- Get the MAC addresses associated with a given IP address over a given time.
- See the entire IP history database as a comma-separated file.
- View the attributes of the lease history (the lease history detail report)—See Querying IP Lease History, on page 40.

You must use additional administrative functions to trim the IP history database of records, to keep the size of the database from growing without bounds.

Note When the state of an existing lease changes (for example, when it is configured as a reserved IP address or it is deactivated), the change does not appear as a lease history change at the regional. With detail collection disabled, a lease history change appears only when the lease transitions from leased to not leased or is assigned to another client.

Enabling Lease History Recording at the Local Cluster

You must explicitly enable lease history recording for the local cluster DHCP server. The DHCP server logs IP history recording errors in the usual DHCP log files.

When the lease history is enabled on a local cluster it impacts the performance of the server and the size of the lease state database. A history record is created for the lease whenever a lease ends (expires or is released); a lease that a client renews over a long period does not cause a history record to be created. The size of each lease history record depends on many factors, but a good estimate is about 1 KB per record. Depending on the rate at which the lease ends and the duration over which lease history is kept, this could result in a sizeable number of lease history records being created and thus requires a considerable disk space. This could be many orders larger than the space needed for the active leases.

Local Advanced Web UI

To enable lease history recording, do the following:

Step 1 From the **Deploy** menu, choose **DHCP Server** under the **DHCP** submenu to open the Manage DHCP Server page.

Step 2 Click the **Local DHCP Server** on the DHCP Server pane.

Step 3 On the Edit Local DHCP Server page, look for the Lease History attributes:

- *Lease History (ip-history)*—Enable or disable the lease history database for v4-only (DHCPv4), v6-only (DHCPv6), or both.
- *ip-history-max-age*—Maximum age of the lease history to collect. With lease history set to v4 only, v6 only, or both the DHCP server periodically examines the lease history records and deletes any records with lease history bindings older than this age threshold.

Step 4 Click **Save**.

Step 5 Reload the server.

CLI Commands

To enable lease history recording, you must explicitly enable recording IP (lease) history for IP addresses by using **dhcp set ip-history=<value> (v4-only, v6-only, both, or disable)**.

Querying IP Lease History

Once you have leases, you can query for their history. You can query IP lease history either from a local or a regional cluster. Set up the local cluster containing the DHCP server as part of the regional cluster, and enable polling for the lease history data from the regional cluster (see the *"Enabling Lease History Collection" section in Cisco Prime Network Registrar 11.2 Administration Guide*).

You can adjust the polling criteria for the cluster in the regional cluster web UI by using the attributes described in the *"Polling Utilization and Lease History Data" section in Cisco Prime Network Registrar 11.2 Administration Guide*.

You must also set the selection criteria for querying the lease history data, as described in the following sections.

Local Advanced and Regional Advanced Web UI

To query the IPv4 lease history, do the following:

Step 1 From the **Operate** menu, choose **DHCPv4 Lease History** under the **Reports** submenu to open the DHCP Lease History Search page.

Note You can use the Search button in the Local Advanced web UI to move to DHCP Lease Search page. This button helps you to toggle between lease history search page and active leases search page.

Step 2 Choose the Filter attribute and the Type from the drop down lists and enter the value of the filter type selected in the Value field.

Step 3 Click **Search** to display the list of leases.

Local Advanced and Regional Advanced Web UI

To query the IPv6 lease history, do the following:

Step 1 From the **Operate** menu, choose **DHCPv6 Lease History** under the **Reports** submenu to open the DHCP v6 Lease History Search page.

Note You can use the Search button in the Local Advanced web UI to move to DHCP v6 Lease Search page. This button helps you to toggle between lease history search page and active leases search page.

Step 2 Choose the Filter attribute and the Type from the drop down lists and enter the value of the filter type selected in the Value field.

Step 3 Click **Search** to display the list of leases.

Note The regional server only searches its version of the lease history which is as recent as the latest poll. For the most up-to-date data, this might require performing an explicit lease history poll for the regional to retrieve the latest lease history data.

Using the iphist Utility

You can query the IP history database at the local as well as regional clusters and direct the results to standard output or a file by using the **iphist** utility. The default location is:

/opt/nwreg2/local/usrbin

From the command prompt, change to the above location and run the utility using the syntax:

iphist [*options*] {*ipaddr* | **all**} [*start-date* | **start** [*end-date* | **end**]]

The IP address is a single address or the keyword **all**, the start date is in local time or the keyword **start** for the earliest date in the database, and the end date is in local time or the keyword **end** for the last date in the database. However, the output is in Greenwich Mean Time (GMT) by default, unless you use the **–l** option to specify local time.

The full list of command options appears in the table below.

Table 4: iphist Command Options

Option	Description
–N *username*	Administrator username. If omitted, you are prompted for the username.
–P *password*	Administrator password. If omitted, you are prompted for the password.
–C *cluster* [**:***port*]	Destination server and optional SCP port.
–6	Output DHCPv6 leases
–a	Shows the lease attributes, visibility 3.
–f *format*	Format of the output lines. The default format is: **"address,client-mac-addr,binding-start-time,binding-end-time"**
–t	Print format as title line.
–n *namespace*	Specify the namespace for the address.
–o *file*	Sends output to a file.
–l	Displays output in local time rather than the default UTC/GMT.
–i	Displays output for delegated prefix that includes specified IPv6 address (only with –6).
–s {self | partner}	Restricts the leases to the self or partner.
–v	Displays the output version.
–z *debug-args*	Sets the debug output levels.

Dates can use this syntax (quotation marks are required if space characters are included):

- *month* /*day* /*year* @*hour* :*min* :*sec* (for example, 8/28/2007@10:01:15), with the time optional
- *month* /*day* /*year hour* :*min* :*sec* (for example, “8/28/2007 10:01:15”), with the time optional
- *month day hour* :*min* :*sec year* (for example, “Aug 28 10:01:15 2007”), with the seconds optional
- Keywords **start**, **end**, or **now** (for the current time)

The date filtering is intended to limit the output to leases that were active during that time. This means that they can begin before the specified start date, as long as they do not end before the start date. They can also not begin after the specified end date. For example, invoking the command:

```
# ./iphist -N user -P password all “Aug 28 00:00 2008” “Dec 31 23:59:59 2008”
```

for the following leases:

Lease 1	Begin	Jan 01 2008	End	Jun 30 2008
Lease 2	Begin	Mar 10 2008	End	Sep 01 2008
Lease 3	Begin	Jun 01 2008	End	Sep 30 2008

Lease 4	Begin	Jan 01 2009	End	Mar 10 2009

would return just Lease 2 and Lease 3, because they both end after the specified start date of the query, even though they both begin before that date. The other two are out of range, because they either end before the specified start date or begin after the specified end date of the query.

The values on each line depend on the specific lease object that the DHCP server stores. You can specify the values to include using the **iphist –f** *format* command.

The *format* argument is a list of lease attribute names, enclosed in quotation marks with the names separated by commas, that provides the template for the output lines. The default output is *ipaddress*, *client-mac-addr*, *binding-start-time*, *binding-end-time*.

For example:

```
# ./iphist -f "address,client-mac-addr,binding-start-time,binding-end-time" all
```

The output is a sequence of lines terminated with a newline sequence appropriate to the operating system (\n on UNIX). Each line contains data on a single lease record. The format of the lines is generally comma-separated values enclosed in quotation marks. To use a literal backslash (\) or quotation mark (") inside quotation marks, precede each with a single backslash (\). New lines in attributes are printed as **\n**.

The table below lists some of the common lease object attributes you can include in the output. Also, see the help for the **lease** command. To get a full list, use **iphist -a**.

Table 5: IP History Query Output Attributes

Lease Attribute	Description
address	IP address of the lease.
binding-start-time	Start time of the lease binding.
binding-end-time	End time of the lease binding.
client-binary-client-id	Binary form of the client MAC address.
client-dns-name	Latest DNS name of the client known by the DHCP server.
client-domain-name	Domain where the client resides.
client-flags	A number of client flags.
client-host-name	Hostname that the client requested.
client-id	Client ID requested by or synthesized for the client.
client-last-transaction-time	Date and time when the client most recently contacted the server.
client-mac-addr	MAC address that the client presented to the DHCP server.
client-os-type	Operating system of the leased client.
expiration	Date and time when the lease expires.
flags	Either reserved or deactivated.

Lease Attribute	Description
lease-renewal-time	Minimal time that the client is expected to issue a lease renewal.
lease-rebinding-time	Minimal time that the client is expected to issue a rebinding request.
relay-agent-circuit-id	Contents of the *circuit-id* suboption (1).
relay-agent-option	Contents of the option from the most recent client interaction.
relay-agent-remote-id	Contents of the *remote-id* suboption (2).
relay-agent-server-id-override	IP address in the *server-id-override* suboption.
relay-agent-subnet-selection	IP address in the *subnet-selection* suboption.
relay-agent-vpn-id	Contents of the *vpn-id* suboption.
start-time-of-state	Date and time when the lease changed its state.
state	One of available, expired, leased, offered, or unavailable.
vendor-class-id	Vendor class ID requested by the client.
vpn-id	Identifier for the VPN, if any.

Trimming Lease History Data

If you enabled IP history trimming at the regional cluster, the IP history database is automatically trimmed so that you can reclaim disk space. Each history record has an expiration time. Trimming is necessary for the DHCP server itself, as well as for the CCM regional server that polls the DHCP server for history data.

The CCM server performs background trimming at the regional cluster, which trims off the lease history data older than a certain age at regular intervals. The trimming interval is set by default to 24 hours, and the age (how far back to go in time before trimming) to 24 weeks. The DHCP server at the local cluster performs daily automatic trimming (at 3:00 A.M. local time), and stores four weeks of data by default.

Regional Web UI

To trim lease history data, you must be a central configuration administrator:

Step 1 From the **Operate** menu, choose **Manage Servers** under the **Servers** submenu to open the Manage Servers page.

Step 2 Click **CCM** in the Manage Servers pane to open the Edit Local CCM Server page.

Step 3 Under the **Lease History Settings** section, set the following attributes (you can use the **s**, **m**, **h**, **d**, **w**, **m**, or **y** suffix with values you enter):

- *lease-hist-trim-interval*—How often to trim the old lease history data automatically, the default being daily. If set to 0, no automatic lease trimming occurs, which is not recommended due to the increasing disk space used. The bounded values are 0 to one year.
- *lease-hist-trim-age*—Provided that the *lease-hist-trim-interval* is not set to 0, how far back in time to trim the old lease history data automatically, the default being 24 weeks. The bounded values are one day to one year.

Step 4 To force immediate trimming, at the bottom of the page find the Trim/Compact Inputs section (compacting is available only for DHCP utilization data). Set the Trim/Compact age to a desired value. This age is how far in time to go back to trim the lease history data. There are no bounds to this value. However, if you set a very small value (such as 1m), it trims or compacts very recent data, which can be undesirable. In fact, if you set it to zero, you lose all of the collected data. Setting the value too high (such as 10y) may end up not trimming or compacting any data.

Step 5 If you are trimming immediately, click **Trim All Lease History**.

You can adjust the trimming that the DHCP server itself performs by setting the *ip-history-max-age* attribute. If *ip-history* is set, the DHCP server accumulates database records over time as lease bindings change. This parameter establishes a limit on the age of the history records kept in the database. The server periodically examines the lease history records, establishes an age threshold based on this parameter, and deletes any records that represent bindings that ended before the threshold. The preset value is four weeks.

Running Lease Utilization Reports

Lease utilization reports show the current utilization of address blocks, subnets, and scopes. For both user interfaces, see Generating Utilization History Reports.

Local Advanced Web UI

View the current utilization for address blocks, subnets, and scopes from pages in the Address Space function.

CLI Commands

To view lease utilization reports, use **report**.

Receiving Lease Notification

The CLI provides the feature of sending notifications if the number of available IP addresses equals or falls below a certain threshold. The **lease-notification** command specifies, through an *available* attribute, when the notification should occur if the number of available leases reaches or falls below a certain threshold. You can e-mail the report to a user. Although you can use the command interactively, its primary use is in an automated procedure such as a UNIX **cron** task.

The following example sets up lease notification for examplescope for when its free addresses fall to 10%. It sends the report to recipients billy, joe, and jane, on a specific Windows mail host:

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

The output consists of an explanatory header, a table containing a row for each scope in which the number of free addresses is equal to or less than the threshold, and possible warnings related to the scopes and clusters requested.

Cisco Prime Network Registrar uses the default cluster and the .nrconfig file by default, unless you specify otherwise. For the command syntax, see the help for the **lease-notification** command.

Running Lease Notification Automatically

You can run **lease-notification** periodically by means of the **cron(1)** command by supplying **crontab(1)** with the command to run.

This example, specified to **crontab**, runs **lease-notification** at 00:15 and 12:15 (15 minutes after midnight and noon), Monday through Friday (note that this encompasses a single command line):

```
15 0,12 * * 1-5 . .profile; /opt/nwreg2/local/usrbin/nrcmd lease-notification available=10\%
 config=/home/jsmith/.nrconfig addresses=192.32.1.0-192.32.128.0
recipients=jsmith,jdoe@example.com >/dev/null 2>&1
```

You can perform **crontab** editing by running the UNIX **crontab -e** command. Set your EDITOR environment variable before running the command, unless you want to use **ed(1)**. See the **crontab(1)** man page for additional details.

Note that you must supply the full path of the CLI command on the **crontab** command line. You can determine the full path in your environment with the UNIX **which nrcmd** command.

Also, when you run the **lease-notification** *command* by means of **crontab**, the **nrcmd** command ignores the user environment variables CNR_CLUSTER, CNR_NAME, and CNR_PASSWORD. Because other viewers can view the command being run, do not provide the password through the **-P** option on the command line, for security reasons.

Supply the cluster name, user, and password information for the cluster you want the **nrcmd** command to run from in a .profile or other file in the home directory of the user running **crontab -e**. For example:

```
CNR_CLUSTER=host1
export CNR_CLUSTER
CMR_NAME=admin1
export CNR_NAME
CNR_PASSWORD=passwd1
export CNR_PASSWORD
```

The **. .profile** specification in the **crontab** entry explicitly reads the file. The first dot (.) is the shell command that reads the file and you must follow it with at least one space character. For notification on a different cluster (or clusters) than where **nrcmd** is running, specify this information:

- Clusters to check in a config file (see Specifying Configuration Files for Lease Notification, on page 46).
- Fully specified path as in the sample **crontab** entry at the beginning of this section.

You can prevent others from examining or changing the contents of the .profile and the configuration file that you create by changing its permissions with the **chmod go-rwx** *config-file* UNIX command.

Specifying Configuration Files for Lease Notification

If you omit a configuration file, **lease-notification** looks for a default .nrconfig file in your current directory, then in your home directory, and finally in the /var/nwreg2/{local | regional}/conf directory. Cisco Prime Network Registrar uses the first file it encounters. Each line of the file must either begin with the character # (comment), a section header enclosed in square brackets, or a parameter/value pair or its continuation. Cisco Prime Network Registrar strips leading space characters from each line and ignores blank lines.

Dynamic Lease Notification

The DHCPv4 and DHCPv6 dynamic lease notification feature allows an external client application to receive updates about the IP address binding activity of the DHCP server. This feature can be used to update an external database with lease activity or trigger actions, such as lawful intercept, when specific lease activity takes place.

Note Dynamic Lease Notification provides only the current lease state information. It does not guarantee that all the lease state changes are reported. Lease state changes are lost under certain conditions, such as when the connection to the DHCP server is down or congested.

The dynamic lease notification feature extends the DHCP server to support additional capabilities and includes a sample client (written in Java), which demonstrates the features by storing the lease state information into a MySQL database.

Using Dynamic Lease Notification

To use Dynamic Lease Notification:

1. You must create a dhcp-listener object on the local cluster. The dhcp-listener object specifies the port on which the server listens for incoming TCP connections and other attributes for these connections (see DHCP Listener Configuration, on page 52). You must reload the DHCP server after creating the dhcp-listener object.
2. A dynamic lease notification client must establish a TCP connection with the DHCP server and make any of these requests:
   - Bulk leasequery—This request is made to obtain the current state of all leases in the DHCP server that have changed state since a specific point in time. The current state of all leases is sent when no time is specified (or zero is specified for the time). This is similar to the UDP-based DHCPv4 leasequery (RFC 4388) and DHCPv6 leasequery (RFC 5460), except that the DHCP server delivers all leases to the client in response to a single request. Typically, a bulk leasequery is used to initialize an external database. It is also used to bring that database up to date after some interruption of an active leasequery, where the catch-up time was too great for the active leasequery to return the missed data.
   - Active leasequery—This request is made to obtain lease state information for all future significant lease changes that the DHCP server will make. When the DHCP server writes significant lease state information to its database, the lease state information will be sent over the TCP connection.
   - Active leasequery with catchup—This request is made to obtain future lease state changes and the latest data from recently changed leases. It allows the dynamic lease notification client to retrieve the latest data on recently changed leases that were missed during a short period of connection loss, such as during a restart of the dynamic lease notification client or DHCP server. The active leasequery with catchup fetches only the current state of a lease; it does not fetch the data on all intermediate lease state changes that might have been missed.

The DHCP server sends the lease state information to the dynamic lease notification client in a stream of leasequery messages. For a bulk leasequery, the lease state information is sent as soon as the DHCP server has time for processing. For an active leasequery, the lease state information is sent as lease state changes occur. The dynamic lease notification client can process these messages to take appropriate actions such as updating its database.

Note While the DHCP server supports multiple dynamic lease notification clients, it is recommended to keep the number of clients to a minimum as multiple clients can impact the DHCP server's leasing performance.

In a failover configuration, only the active failover partner which interacts with the DHCP client sends dynamic lease notification updates to the dynamic lease notification clients with an active leasequery request. Therefore, to receive complete information, a dynamic lease notification client must connect to both the failover partners.

The server determines whether a lease is queued for active leasequery notifications based on the *leasequery-send-all* attribute of the dhcp-listener. If this attribute is enabled, the DHCP server always sends a notification to an active leasequery client. If this attribute is disabled or unset, the DHCP server only sends notifications which are necessary to maintain accurate state in the active leasequery client.

You can also control the leasequery notifications using extensions. Extensions can decide whether a lease is queued for active leasequery notifications using the *active-leasequery-control* request and response data dictionary items as described in Using Extension Points.

Sample Lease Notification Client

Cisco Prime Network Registrar provides a standalone sample Java client. The standalone sample Java client collects the lease state data from one or more DHCP servers, and updates the SQL database with the most current lease data. The sample Java client is designed to accept lease state updates from both failover partners and ensures that the latest lease state information is in the SQL database (even when updates are received out of order). If you use the sample Java client, it is not necessary to know the complete details of the bulk and active leasequery protocols. The sample Java client sources are provided; thus if the sample Java client does not meet your needs, it is recommended you modify it rather than implementing your own.

The sample Java client performs a bulk leasequery when it connects to a server for the first time to obtain the state of all leases. If the sample Java client has communicated with the server before, it attempts an active leasequery with catchup. The sample Java client performs a bulk leasequery only if the active leasequery with catchup indicates that catch-up data is not available, such as if the client was down for a while or the DHCP server was reloaded.

The sample Java client supports configurations with multiple VPNs and multiple servers. However, the sample Java client assumes that the leases across these servers are unique with respect to VPN and IP address. If two servers lease out the same IP address in a VPN or global namespace, the SQL database will contain a record of only one of the two leases. This does not apply to failover pairs, but rather to two independent DHCP servers. The sample Java client must also be configured to communicate with both the failover partners of a failover-pair to keep the SQL database up-to-date.

Note The sample Java client is available at *install-path*/examples/dhcp/cnrnotify.jar. A text readme file named cnrnotify-readme.txt file is also provided in that directory and must be read first.

The examples/dhcp/cnrnotify.jar is a zip file, which contains:

- The sample Java client source code and Javadoc documentation.
- For example lnc.properties and lnc6.properties files. (Run the client with -listprops option for details on the available properties.)
- The Bulk and Active Leasequery Internet Drafts for the Cisco Prime Network Registrar implementation.
- A document that details the message values, option codes, and vendor-specific data used for Cisco Prime Network Registrar proprietary lease information. As the Internet Assigned Numbers Authority (IANA) has not yet assigned values to the messages and option codes used by the Bulk and Active Leasequery Internet Drafts, this document describes the values that are used in the Cisco Prime Network Registrar.

To extract these items, open the cnrnotify.jar file using a zip tool such as Winzip. (See the cnrnotify-readme.txt file.) To extract the Javadoc, we recommend you use:

```
jar xvf cnrnotify.jar docs_notify
```

The above command is used to extract the documentation.

DHCPv4 Sub-sub Option Codes

Following table lists the sub-sub option codes used while requesting for DHCPv4 leasequery. These codes are present in the cnrnotify-protocol-numbers.txt file which is available in the cnrnotify.jar zip file.

Table 6: DHCPv4 Sub-sub Option Codes

Sub-sub Option Code	Option Name	Option Type
1	oro	one or more bytes of sub-sub option numbers
2	dhcp-state	byte
3	data-source	byte
4	start-time-of-state	duration in past from base-time
5	base-time	absolute time (secs from 1970)
8	client-class-name	string (without zero termination)
9	partner-last-transaction-time	duration in past from base-time
10 0xa	client-creation-time	duration in past from base-time
11 0xb	limitation-id	blob containing limitation-id
12 0xc	binding-start-time	duration in past from base-time
13 0xd	binding-end-time	negative/positive value representing duration in future/past from base-time
14 0xe	fwd-dns-config-name	string (without zero termination)
15 0xf	rev-dns-config-name	string (without zero termination)
16 0x10	client-override-client-id	blob containing client-id for client
17 0x11	user-defined-data	string (without zero termination)
18 0x12	scope-name	string (without zero termination)
19 0x13	failover-state-serial-number	4 byte integer, network order
20 0x14	reservation-key	blob, starting with type byte: • 0x2e, 46: string without zero termination • 0x7, 7: blob
21 0x15	client-prl	client's parameter request list, blob of DHCPv4 option code

DHCPv6 Sub-sub Option Codes

Following table lists the sub-sub option codes used while requesting for DHCPv6 leasequery. These codes are also present in the cnrnotify-protocol6-numbers.txt file which is available in the cnrnotify.jar zip file.

Table 7: DHCPv6 Sub-sub Option Codes

Sub-sub Option Code	Option Name	Option Type
1	oro	one or more bytes of sub-sub option numbers
2	dhcp-state	byte
3	data-source	byte
4	start-time-of-state	duration in past from base-time
5	base-time	absolute time (secs from 1970)
8	client-class-name	string (without zero termination)
9	partner-last-transaction-time	duration in past from base-time
10 0xa	client-creation-time	duration in past from base-time
12 0xc	binding-start-time	duration in past from base-time
13 0xd	binding-end-time	negative/positive value representing duration in future/past from base-time
14 0xe	fwd-dns-config-name	string (without zero termination)
15 0xf	rev-dns-config-name	string (without zero termination)
16 0x10	lookup-key	blob containing client-id for client
17 0x11	user-defined-data	string (without zero termination)
18 0x12	prefix-name	string (without zero termination)
19 0x13	failover-state-serial-number	4 byte integer, network order
20 0x14	reservation-key	blob, starting with type byte: • 0x2e, 46: string without zero termination • 0x7, 7: blob
21 0x15	failover-partner-lifetime	negative/positive value representing duration in future/past from base-time
22 0x16	failover-next-partner-lifetime	negative/positive value representing duration in future/past from base-time
23 0x17	failover-expiration-time	negative/positive value representing duration in future/past from base-time

Sub-sub Option Code	Option Name	Option Type
24 0x18	client-oro	client's ORO, blob of DHCPv6 two byte option codes

Requirements for Sample Java Client

The requirements for the sample Java client are:

- JDK 11
- The java.sql package from JDK 11
- Installation of a JDBC driver and a compatible database. A specific table (that contains a pre-defined set of columns) must exist in the database.

Tip If the tables do not exist, run the client with -c option. The tables are thus created.

The requirements for MySQL are:

- The latest version of MySQL server.
- The JDBC connector for MySQL.
- The log4j package for logging the sample Java client status and errors.

Note We recommend that you use MySQL-8.0.29 database, mysql-connector-java-8.0.29.jar, log4j-api-2.17.2.jar, and log4j-core-2.17.2.jar.

Once extracted and the lnc.properties file is configured, the sample Java client can be run using:

Step 1 Place all four .jar files (cnrnortify.jar, mysql-connector-java-8.0.29.jar, log4j-api-2.17.2.jar, and log4j-core-2.17.2.jar) in the same directory.

Step 2 Extract lnc.properties/lnc6.properties file in same directory:

For DHCPv4 client:

```
jar xvf cnrnotify.jar com/cisco/cnr/notify/lnc.properties
```

For DHCPv6 client:

```
jar xvf cnrnotify.jar com/cisco/cnr/notify/lnc6.properties
```

Step 3 Configure lnc.properties/lnc6.properties file.

Step 4 Assuming Java executable directory is in the current path, the sample client is run by:

For DHCPv4:

```
java -cp .:cnrnotify.jar:mysql-connector-java-8.0.29.jar:log4j-api-2.17.2.jar:log4j-core-2.17.2.jar
 com/cisco/cnr/notify/LeaseNotificationClient
```

For DHCPv6:

```
java -cp .:cnrnotify.jar:mysql-connector-java-8.0.29.jar:log4j-api-2.17.2.jar:log4j-core-2.17.2.jar
 com/cisco/cnr/notify/LeaseNotificationClient6
```

Local Web UI

The web UI displays and manages the configuration attributes, and displays the related servers' information. The statistics about the lease queries are available on the DHCP Server Statistics page.

Step 1 From the **Deploy** menu, choose **DHCP Server** under the **DHCP** submenu to open the Manage DHCP Server page.

Step 2 Click the **Statistics** tab to open the DHCP Server Statistics page.

The Server Statistics details are displayed in this page.

CLI Command

The existing **dhcp getRelatedServers** command is extended to supply information about the DHCP listeners and any active connections.

```
nrcmd> dhcp getrelatedservers
```

Note You can use this command only on a local cluster.

DHCP Listener Configuration

Using DHCP Listener Configuration, you can configure objects to enable active and bulk leasequery to the DHCP server over TCP connections. A single object is sufficient, unless you want the DHCP server to support listening for connections on multiple TCP ports or need to restrict the addresses on which the server will accept incoming connections.

Local Advanced Web UI

Step 1 From the **Deploy** menu, choose **Listeners** under the **DHCP** submenu, to open the List/Add DHCP TCP Listener page.

Step 2 Click the **Add Listeners** icon in the Listeners pane, enter a name in the name field, then click **Add TCP Listener**.

Step 3 Enter an IP address in the address/ip6address field, to restrict the interface over which the server will accept connections. This is usually unspecified. If you want to configure a IPv6 listener, then enter ip6address. If both address and ip6address are not specified, then the IPv4 address 0.0.0.0 is used.

To restrict the address on which TCP connections are accepted, enter the address in the address (for IPv4) or ip6address (for IPv6) attribute. If no value is entered in either attribute, IPv4 connections to any IPv4 address of the host are accepted. To specify connections over IPv6, you must enter a value in the ip6address attribute (0::0 can be used to accept connections to any IPv6 address of the host). You can only enter a value in either, not both, attributes.

Note You cannot specify both IPv4 and IPv6 listeners for a DHCP server.

Step 4 Enter a value for the port in the port field, if the default value is not appropriate. The default port is the server-port for DHCPv4 and DHCPv6-server-port for DHCPv6.

Step 5 For the *enable* attribute, click true or false radio button. The default value is true.

Step 6 Enter a value for *max-connections*, if the default value 10 is not appropriate.

Step 7 Enter a value for *leasequery-backlog-time*, if the default value 120 is not appropriate.

Step 8 For *leasequery-send-all* attribute, click true or false radio button. The default value is false.

Step 9 Click **Save**.

CLI Commands

The DHCP Listener commands are shown in the table below.

Table 8: DHCP Listener Commands

Action	Command
Create	**dhcp-listener** *name* **create** [*attribute=value*]
Delete	**dhcp-listener** *name* **delete**
List	**dhcp-listener list**
List the names	**dhcp-listener listnames**
Show	**dhcp-listener** *name* **show**
Set	**dhcp-listener** *name* **set** *attribute=value* [*attribute=value ...*]
Get	**dhcp-listener** *name* **get** *attribute*
Unset	**dhcp-listener** *name* **unset** *attribute*
Enable	**dhcp-listener** *name* **enable** *attribute*
Disable	**dhcp-listener** *name* **disable** *attribute*

Lease History Database Compression Utility

The **cnr_leasehist_compress** utility was added to Cisco Prime Network Registrar to compress regional cluster (DHCPv4) lease history databases. This utility does not compress the data directly in the databases, but copies the existing data into new databases that are optimized to be as compact as possible. You can download this utility from the Cisco Prime Network Registrar download section on the Cisco website.

Caution Use the **cnr_leasehist_compress** utility only with the regional cluster lease history database, and when you suspect that the database grew significantly, particularly because of DHCPRELEASE packets.

During the copy operation, you can use this utility to:

- Trim records that are older than a certain interval of time—You would typically use the **–t** option. The interval that you specify with this option uses the Network Registrar time interval format; for example, **30d** for 30 days or **1y** for 1 year.
- Merge records that belong to the same lease and client—You use the **cnr_leasehist_compress** utility to merge records that belong to clients who have reclaimed the lease on an IP address after releasing it. You would typically use the **–m** option. The interval that you specify with this option uses the Network Registrar time interval format; for example, **120s** for 120 seconds or **2m** for 2 minutes.

  While merging records, the utility also corrects lease history records that were terminated abruptly or have an incorrect binding end time (that may have resulted from a subsequent lease operation). This option of merging records also addresses the vast number of records that are created by certain router configurations that introduce an additional load on the servers.

Before you run the **cnr_leasehist_compress** utility:

- Stop the Network Registrar regional cluster; it does not operate on an active regional cluster database.
- Note that you can use it to compress existing lease history data alone; it does not alter how the regional cluster collects future lease history records. If you suspect chatty clients, ensure that the DHCP server does not process DHCPRELEASE messages, because this results in rapid growth of lease history data. In such instances, you may need to run the utility periodically.
- Note that you can use it if you are a service provider and suspect that the regional lease history in your network grew because some devices have known issues, such as repeatedly generating a sequence of DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK, and after 30 seconds, DHCPRELEASE messages. You can choose to drop all DHCPRELEASE messages or those that belong to clients that exceed a configured threshold.
- Note that it writes the new database in the most optimal manner. The new database can initially grow at a considerable rate, but it tapers back to normal after the additional lease history records are collected.

General Comments on Running cnr_leasehist_compress

Caution Follow every step in this procedure carefully. If you skip any step, you might lose lease history data. Note the lease history database that each task involves. Depending on the number of lease history records and the time taken to trim or merge the records, this utility may take several hours or days to run. You can interrupt the utility while it is running if the server reboots before the run is completed. You can resume it later; however, you must specify the same options that you have used in the previous run.

The *install-path* is the path in which you install Cisco Prime Network Registrar.

The table below describes the qualifying options for the **cnr_leasehist_compress** utility.

Table 9: cnr_leasehist_compress Options

Option	Description
–a	Appends all the lease history records in the temporary active database to those in the new database.
–c *limit*	Generates a report when more than a specified number of records (*limit*) merged for a client. When used with the **–f** option, these records are transferred to a log file.

Option	Description
–C	Compresses lease records on write (for details, see CCM's *lease-hist-compression* attribute.
–d *path*	Specifies the path to a new destination database that contains the compressed lease history records.
–e *attrlist*	Overwrites the excluded merge attribute list.
–f *file*	Redirects most lease history record warnings to a log file.
–g	Uses the *dbtxn-seq* and *dbtxn-generation* attributes to generate a new sequence in the numbers that are assigned to all lease history records, which are written into the destination database.
–i *ipaddr*	Transfers the records of a particular IP address to a log file.
–l *limit*	Purges log files after the database reaches the preset limit of 20 files.
–m *time-int*	Merges lease records where the **binding-start-time** of a particular lease falls in the duration of the **binding-end-time** of the previous lease. The recommended value for this option is **120s**.
–n	Compares adjacent records without merging them.
–p	Drops detailed lease history records. You can use this option only if you have enabled detailed lease history. **Note** Cisco Prime Network Registrar no longer supports detailed lease history. However, this option is retained in case of upgrades from a version that supported detailed lease history.
–q *records*	Sets an interval for a periodic progress report that is generated while the utility runs. The default value is **100000**. For example: `+00:00:18 Read 100000 records (0 bad); trimmed 6717; merged 73370; 19912 written (19.91%)`
–r *records*	Limits the number of records that are read from the source database.
–s *path*	Specifies the source database from where the data is copied to a new database.
–t *age*	Specifies a value for trimming records that are older than a certain interval of time. Use the standard Network Registrar time interval for this option, such as **1y** for 1 year or **30d** for 30 days.
–v	Emits the version and exits.
–w *records*	Limits the number of records that are written into the destination database.
–y "*line attr*"	Alters the width of the dump of lease history records. This option is not recommended; however, you can use the value **132 30** for a 132-column output.
–z *{letters}=level*	Debugs the database, specified by using the standard Network Registrar debug tracing syntax.

Running Compression

To run the **cnr_leasehist_compress** utility, do the following:

Step 1 Add *install-path*/lib to the LD_LIBRARY_PATH to provide the utility with access to the Network Registrar libraries:

```
$ bash
# export LD_LIBRARY_PATH=install-path/lib:$LD_LIBRARY_PATH
```

Step 2 Stop the Network Registrar regional cluster:

```
# systemctl stop nwregregional
```

Step 3 Rename the original *install-path*/data/leasehist directory as *install-path*/data/oldleasehist. The /leasehist directory becomes the original database:

```
# mv install-path/data/leasehist
# install-path/data/oldleasehist
```

Step 4 Create a new leasehist directory, which becomes the temporary active database:

```
# mkdir install-path/data/leasehist
```

Step 5 Run the **cnr_leasehist_compress** utility to allow the regional cluster to resume activity:

```
# install-path/bin/cnr_leasehist_compress
> -r 0
> -s install-path/data/oldleasehist
> -d install-path/data/leasehist
> -p
```

Caution Running these commands does not compress the original database. The **–r 0** option is critical as it instructs the utility to create the temporary active database. The regional cluster remains active while the utility compresses the original database.

Step 6 Restart the Network Registrar regional cluster.

```
# systemctl start nwregregional
```

You cannot, however, obtain lease history data from the original database at this time. The regional cluster collects new lease history data and transfers it to the temporary active database. The utility then merges the new lease history data into the new database.

Step 7 Create a new directory called *install-path*/data/newleasehist. This /newleasehist directory becomes the new lease history database:

```
# mkdir install-path/data/newleasehist
```

Tip After the regional cluster populates the new database, you can optionally create this new directory on a different partition and copy it to the final location.

Step 8 Run the **cnr_leasehist_compress** utility to trim, merge, and compress the original database into the new database:

```
# install-path/bin/cnr_leasehist_compress
> -s install-path/data/oldleasehist
> -d install-path/data/newleasehist
> -t trim-time-interval
> -m merge-time-interval
> -f /tmp/cnr-compress.log
```

If the original database contains any detailed lease history records, you must use the **–p** option to acknowledge that it is acceptable for the utility to not transfer these records into the new database. Otherwise, the utility does not run.

Note Cisco Prime Network Registrar no longer supports detailed lease history. However, this option is retained in case of upgrades from a version that supported detailed lease history.

Step 9 Perform the following tasks to append any fresh lease history records to the new database after the utility processes the entire original database.

Note Do not restart the regional cluster until you have completed the following procedure. If the system reboots during the following procedure, repeat these steps.

a) Stop the Network Registrar regional cluster:

```
# systemctl stop nwregregional
```

b) Run the **cnr_leasehist_compress** utility to append new lease history records to the new database:

```
# install-path/bin/cnr_leasehist_compress
> -a
> -s install-path/data/leasehist
> -d install-path/data/newleasehist
> -m merge-time-interval
> -f /tmp/cnr-append.log
```

Caution The **–a** option is critical as it indicates that the utility should append the lease history records in the temporary active database to those in the new database. We recommend that you use the same *merge-time-interval* value that you used for the original database.

c) After the utility completes the task of appending the newly collected lease history records, rename the temporary active database directory, *install-path*/data/leasehist, as *install-path*/data/tmpleasehist:

```
# mv install-path/data/leasehist
# install-path/data/tmpleasehist
```

d) Rename the new database directory, *install-path*/data/newleasehist, as *install-path*/data/leasehist:

```
# mv install-path/data/newleasehist
# install-path/data/leasehist
```

Step 10 Start the Network Registrar regional cluster:

```
# systemctl start nwregregional
```

Step 11 Verify the regional lease history data by using the Network Registrar web UI.

Step 12 Archive the original database, in *install-path*/data/oldleasehist, and the temporary active database, in *install-path*/data/tmpleasehist. Ensure that you include all subdirectories and files when you archive the database.

Step 13 Delete the original database and temporary active database:

```
# rm -rf install-path/data/oldleasehist
# rm -rf install-path/data/tmpleasehist
```

Elastic Lease Times

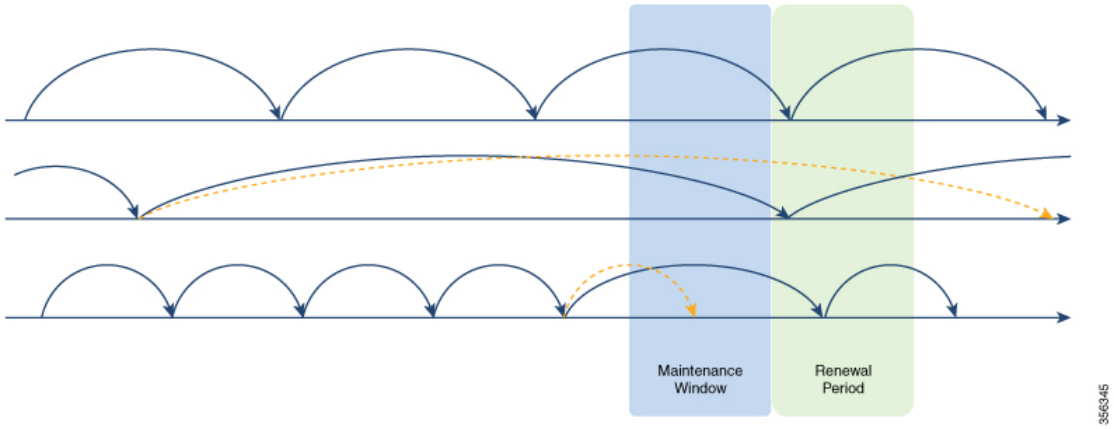
You may need to reconfigure the network, because of the need to renumber certain network segments or to make the configuration change effective quickly. Typically, this has been done by reducing the lease times for clients in advance of the change, then applying the change, and restoring the lease times to their original values. In other words, you need to compress the renewal times into a relatively narrow window (maintenance window), and then expand them back to even out the load on the server. These steps are manual and error prone. Cisco Prime Network Registrar helps to automate this process and to reduce the renewal load on the DHCP server before, during, and after the maintenance window.

Scheduling a Network Reconfiguration

In Cisco Prime Network Registrar, you can schedule the maintenance window, so that it is less prone to errors and forgetting to reset lease times. You can configure the start time, end time, and refresh period of the desired maintenance window. Also, you can specify whether the maintenance window is server wide or applies only to specific scopes, links, or prefixes. The server tries to adjust lease, renewal (T1), and rebinding (T2) times to avoid the client's attempt to contact the DHCP server between the start and end time of the maintenance window, as the server may shutdown during this period. During the maintenance window, the DHCP server uses the minimum lease time and after the maintenance window, it ramps up lease time to original, but keeps the renewals spread out. The renewals during and after the reconfiguration are well distributed to minimize any server load spikes. You can eventually remove the maintenance window configuration or replace with a new configuration. The server ignores the maintenance that occurred in the past.

Figure 1: Maintenance Window, on page 58 shows three clients with different lease times, interacting with the maintenance window. In the first (top) case, there is no change as client is coming in during the renewal period. In the second (middle) case, the server shortens the times to assure the client renews during the renewal period. In the third (bottom) case, the server increases the times to avoid client renewal during the maintenance window (as server may not be reachable).

Figure 1: Maintenance Window



You can create, edit, and delete a single maintenance window, which means that, there can be only one possible maintenance window at a time in the server. The DHCP server will load the maintenance window if configured. It will ignore to load the maintenance window if the current time is after the end time plus the renewal period (time interval after the end of the maintenance window by which all clients should have the updated configuration). Also, it will not load if the distributing of lease renewals is not enabled (see Distributing Lease

Renewals, on page 60). For scopes, links, or prefixes to which the maintenance window applies, the server will alter the lease and/or renewal times sent to clients as follows:

- Any lease time given to a client before the end of the maintenance window will not exceed the end time of the maintenance window plus the renewal period.
- Any renewal time given to a client before the end of the maintenance window will not exceed the end time of the maintenance window plus 1/2 the renewal period.
- Any lease time given to a client that ends between the start and end time of the maintenance window will be adjusted to expire somewhere between the end of the maintenance window plus 1/2 the time interval after and before the end time plus the interval after.
- Any renewal time given to a client that occurs between the start and end time of the maintenance window will be adjusted to trigger a renewal somewhere between the end of the maintenance window and 1/2 of the renewal period.

Note Failover time restrictions continue to apply and are not altered because of the maintenance window. These restrictions may prevent the server from optimizing the lease, renewal (T1), and rebinding (T2) times for some clients.

Adding a Maintenance Window Object

To add a maintenance window object, do the following:

Local Advanced Web UI

Step 1 From the **Deploy** menu, choose **Maintenance Window** under the **DHCP** submenu. This opens the List/Add Maintenance Windows page.

Step 2 Click the **Add Maintenance Window** icon in the left pane and enter the details in the following fields:

- **Name**—Name of the DHCP maintenance window object.
- **Start Date**—The date and time at which the maintenance window starts. This is when the DHCP server is expected to be stopped.
- **End Date**—The date and time at which the maintenance window ends. This when the DHCP server is expected to be available again (after any configuration changes have been made).
- **Refresh Period**—The period after the end of the maintenance window when all the affected clients should contact the server to pick up any newly configured information.

Step 3 Click **Add Maintenance Window**.

Step 4 If you want to apply the maintenance window to specific scopes, prefixes, or links, do the following:

- The scopes that have their *maintenance* attribute (in the List/Add DHCP Scopes page) set to enabled are listed under the Scopes area. To apply the maintenance window to a specific scope, click the **disabled** radio button next to the **Configure Scopes** option, and then select or add the required scopes from the Scopes area. If you click the **enabled** radio button, all the scopes in the configuration will participate in the current maintenance window.

- The prefixes/links that have their *maintenance* attribute (in the List/Add DHCP v6 Prefixes or List/Add DHCP v6 Links page) set to enabled are listed under the Links or Prefixes area. To apply the maintenance window to a specific prefix or link, click the **disabled** radio button next to the **Configure Prefixes/Links** option, and then select or add the required links or prefixes from the Links or Prefixes area respectively. If you click the **enabled** radio button, all the prefixes and links in the configuration will participate in the current maintenance window.

Step 5 Click **Save**.

You can edit the details of the maintenance window object in the Edit Maintenance Window page. To delete the maintenance window object, select the name of the maintenance window object in the left pane, click the **Delete Selected Maintenance Window** icon in the left pane, and then confirm the deletion.

Note Deleting the maintenance window object also clears all the scope, prefix, and link maintenance attributes.

CLI Commands

Use the **dhcp-maintenance-window** *name* **create** [*attribute=value ...*] command to create a maintenance window object. Use the **dhcp-maintenance-window** *name* **delete** command to delete the maintenance window object.

Use the **dhcp-maintenance-window clearMaintenance** [**dhcpv4** | **dhcpv6**] to clear the maintenance flags on all scopes and/or all prefixes/links. If **dhcpv4** is specified, only scopes are cleared. If **dhcpv6** is specified, only prefixes/links are cleared. If neither is specified, all are cleared.

For a complete list of all maintenance window commands, see the **dhcp-maintenance-window** command in the CLIGuide.html file in the /docs directory or use **help dhcp-maintenance-window** in the CLI.

Distributing Lease Renewals

The DHCP server adjusts the client renewals to ensure that the lease renewal load is as evenly distributed as possible and thus to avoid spikes in renewal traffic. Renewal traffic spikes can occur after a maintenance window, network (or power) outage where a large number of clients return at once. This feature is enabled by default to avoid these spikes.

The server maintains the count of the number of clients renewing within a bucket interval. When the server determines a renewal time for a client (50% of lease time), it checks whether the value of that bucket exceeds the norm (number of clients / (latest renewal time-current time / bucket interval) rounded up). If the norm is exceeded, the server picks a random value between 20% and 120% of the renewal time, and will check that bucket against the norm. This process is repeated a limited number of times until a bucket is found that is below the norm, or if not, the time for the bucket with the lowest count is used.

Note The server does not adjust the count if a bucket is less than 10 renewals/second, as the server can easily handle that load.

Figure 2: Distribute Lease Renewals Example

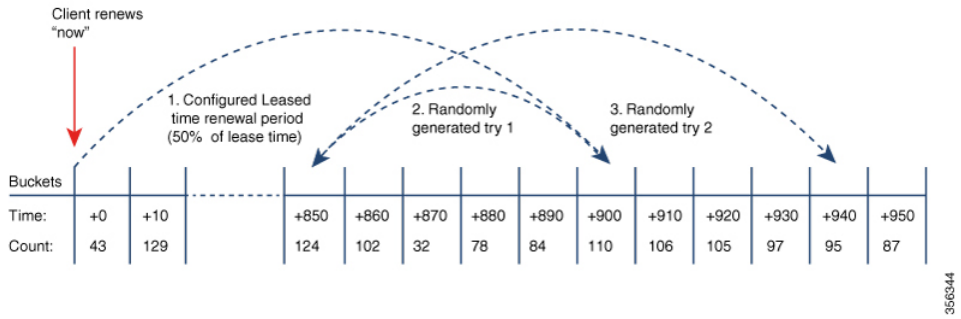


Figure 2: Distribute Lease Renewals Example, on page 61 shows an example of the distribute renewals feature. In this example, the server adjusts renewal time if bucket at client's normal renewal time (50% of lease time of 1800 seconds = 900 seconds) exceeds the threshold of expected clients renewing in that bucket's period. Here, the server picks a random alternative renewal time (between 20% to 120% of original renewal). But, first try also exceeds threshold, so a secondary try is attempted, which finds renewal time (944) that is in a bucket which is below the threshold. Client is given that renewal time (944 seconds).

For DHCPv4, when this feature is enabled, the server will force sending the *dhcp-renewal-time* option (58) and the *dhcp-rebinding-time* option (59). For DHCPv6, the server always sets the T1/T2 fields in the IA_NA and IA_PD options, so there is no impact to that processing.

Controlling the Distribute Renewals Feature

To control the distribute renewals feature, do the following:

Local Advanced Web UI

Step 1 From the **Operate** menu, choose **Manage Servers** under the **Servers** submenu to open the Manage Servers page.

Step 2 Click **DHCP** in the Manage Servers pane to open the Edit Local DHCP Server page.

Step 3 Under the **Distributed Renewals** section, set the following attributes:

- *distribute-renewals*—Controls whether the DHCP server is allowed to adjust the renewal time to smooth the server's renewal load.

  Note If the *dhcp-lease-time* option (51) or preferred lifetime for any configured policy is set to be longer than 180 days, the server will not enable this feature.

- *distribute-renewals-max-renewal-time*—Controls the maximum renewal time used by the server in adjusting renewals to smooth the server's renewal load. If this attribute is unset (or 0), the server will determine this based on 50% of the *dhcp-lease-time* option (51) or preferred lifetime across all named and embedded policies.
- *distribute-renewals-bucket-interval*—Controls the time interval for the buckets used to smooth the server's load. If this attribute is unset, the server uses 10 seconds unless the number of buckets would exceed 100,000; in which case, the server will use a time interval to limit the buckets to at most 100,000.

Step 4 Click **Save**.

CLI Command

To disable the distribute-renewals feature, use **dhcp disable distribute-renewals**. To enable the distribute-renewals feature, use **dhcp enable distribute-renewals**. You can also use the **dhcp set** commands to alter the *distribute-renewals-max-renewal-time* and *distribute-renewals-bucket-interval* values.

Viewing DHCP Renewal Report

The DHCP Renewal Report page on the local web UI displays the expected load of renewals on the DHCP server in a graphical manner. This shows the number of clients that are expected to renew in the future in a specific time interval (bucket).

You can also see the renewal data from the web UI's dashboard. For more information, see DHCP Renewal Data.

To view the DHCP renewal report, do the following:

Local Web UI

Step 1 From the **Operate** menu, choose **Manage Servers** under the **Servers** submenu to open the Manage Servers page.

Step 2 Click **DHCP** in the Manage Servers pane to open the Edit Local DHCP Server page.

Step 3 Click the **DHCP Renewal Report** tab.

Step 4 Enter the desired number of buckets in the **Number of Buckets** field. This specifies the number of buckets into which the renewal data is reported. A bucket represents the clients expected to renew during that time interval.

Step 5 Click **Show**.

The DHCP renewal data is displayed in the form of graph, with number of clients renewing within specific intervals along the Y-Axis and date/time stamp along the X-Axis.

CLI Command

Use **dhcp getRenewalData** [*max-buckets*] to report the information related to the distribute renewals feature. By default, the expected number of client renewals over time is shown in at most 20 buckets, but this can be overriden by specifying the desired number.

This will display some information about the configuration and also a (character cell) graph of the number of clients in each renewal bucket.