CHAPTER 22

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.

See Also

Configuring Leases in Scopes
Searching Server-Wide for Leases, page 22-9
Using Client Reservations, page 22-12
Creating Lease Reservations, page 22-14
Setting Advanced Lease and Reservation Properties, page 22-16
Running Address and Lease Reports, page 22-22
Querying Leases, page 22-31
Dynamic Lease Notification, page 22-38
Sample Lease Notification Client, page 22-39

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.

See Also

Viewing Leases, page 22-2
Lease States, page 22-2
Guidelines for Lease Times, page 22-3
Importing and Exporting Lease Data, page 22-5
Pinging Hosts Before Offering Addresses, page 22-7
Deactivating Leases, page 22-8
Excluding Leases from Ranges, page 22-8

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 *Quick Start Guide for Cisco Network Registrar* or the "Defining and Configuring Scopes" section on page 20-3, then wait for the DHCP server to generate leases based on these addresses.

Local Basic Web UI

To view leases, choose **Scopes** from the **DHCP** menu to open the Manage Scopes page, then click the View icon (🔍) in the Leases column for the scope. This opens the List Leases for Scope page, where you can click each lease to manage it.

See the "Lease States" section on page 22-2 for a description of the values in the State column. For guidelines as to the lease expiration time, see the "Guidelines for Lease Times" section on page 22-3.

To open the Manage DHCP Lease page, click the lease IP address.

Local Advanced Web UI

From the **DHCP** menu, choose **Scopes** to open the List/Add DHCP Scopes page. You can then click the View icon (🔍) in the Leases column for the scope; or you can click the name of the scope to open the Edit DHCP Scope page, then click **List Leases** in the Leases area of the page. Both actions open the List DHCP Leases for Scope page, where you can manage the leases as in Basic mode.

CLI Commands

Use **lease** *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** *addr* **macaddr** command shows the MAC address of the lease, whether or not that lease is reserved or active. The **lease** *addr* **list –macaddr** 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** *addr* **list –subnet network** *netaddr netmask*.

Lease States

A lease can be in one of the states described in Table 22-1.

Table 22-1 Lease States

State	Description
Available	IP address available to be leased.
Unavailable	Not leasable. See the "Handling Leases Marked as Unavailable" section on page 22-21 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.

Table 22-1 Lease States (continued)

State	Description
Deactivated	Not renewable or leasable after the lease expires. See the "Deactivating Leases" section on page 22-8.
Pending available	Failover-related. See Chapter 27, "Configuring DHCP Failover."

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 the "Setting DHCP Request and Response Packet Buffers" section on page 27-20).

- 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 the "Deferring Lease Extensions" section on page 23-8.

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.
- Detailed lease history is disabled if the limiting retention feature is enabled. This is not an issue if detailed lease history has not been used.

- 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.

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. Example 22-1 shows part of a lease data export from a Cisco Network Registrar DHCP server. The blank lines between records appear in the example for clarity; they are not in the actual output.

Example 22-1 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-before-offer* attribute). 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-before-offer* attribute at the DHCP server.

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.

Local Basic Web UI

To deactivate a lease, click the address of the lease on the List Leases for Scope page (see the "Viewing Leases" section on page 22-2). On the Manage DHCP Lease page, click **Deactivate**. The lease now shows as deactivated. To reactivate the lease, click **Activate**.

Local Advanced Web UI

To deactivate a lease, the same operations exist as in Basic mode, except that you click the address of the lease on the List DHCP Leases for Scope page, which opens the Manage DHCP Lease page.

CLI Commands

To deactivate a lease, use **lease** *ipaddr* **deactivate**. To reactivate a lease, use **lease** *ipaddr* **activate**.

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 the "Deactivating Leases" section on page 22-8, 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:

Step 1 From the **DHCP** menu, choose **Scopes** to open the Manage Scopes (Address Pools) page.

Step 2 Click the name of the scope to open the Edit DHCP Scope (Address Pool) 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 Modify 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** *ipaddr* **deactivate**), then remove the range of just that IP address (**scope** *name* **removeRange**). 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
```

Searching Server-Wide for Leases

Using Cisco 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 Network Registrar.

Local Advanced Web UI

To search for DHCPv4 leases, do the following:

Step 1 From the **DHCPv4** menu, choose **Search** to open the DHCP v4 Lease Search page.

You can also go to the DHCP v4 Lease Search page if you choose **Lease History** from the **Address Space** menu. If you choose **Lease History** from the **Address Space** menu, the DHCP v4 Lease History Search page is displayed. You have to click the DHCP v4 Lease Search button to go to the DHCP v4 Lease Search page.

Note You can open the DHCP v4 Lease Search page by clicking the DHCP v4 Lease Search button in the DHCP v4 Lease History Search page (choose Lease History from the Address Space v4 menu to open the DHCP v4 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 Table 5-4 on page 5-36).

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.

To search for DHCPv6 leases, do the following:

Step 1 From the **DHCPv6** menu, choose **Search** to open the DHCP v6 Lease Search page.

You can also go to the DHCP v6 Lease Search page if you choose **Lease History** from the **Address Space v6** menu. If you choose **Lease History** from the **Address Space v6** menu, the DHCP v6 Lease History Search page is displayed. You have to click the DHCP v6 Lease 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—IP address From value to an IP address To value.
- Regular Expression—Value is a Regular Expression in regex syntax. (For common regex usage, see Table 5-4 on page 5-36).

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 DHCPv6 space, use the following **lease6 list** syntax:

```
nrcmd> lease6 list
    [-duid=client-id]
    [-lookup-key=key] [-blob | -string]]
    [-macaddr=mac-addr]
    [-cm-macadd=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 -macaddr=01:02:03:04:05:06
nrcmd> lease6 -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 Table C-4 on page C-7 for details on these suboptions.)

Using Client Reservations

In Cisco Network Registrar versions earlier than 7.2, the only option for clients to get the lease they want was to create a lease reservation (see Creating Lease Reservations, page 22-14). 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 Network Registrar reservations with databases is very complex. The client reservation feature helps in reducing this complexity.

The current functionality supported by Cisco 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 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 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 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 Network Registrar can look up clients in an LDAP repository (external to Cisco Network Registrar) and these clients may specify client reservations.
- Using extensions—Cisco Network Registrar can be set up to communicate with external servers or databases using extensions.

The client entries, maintained either within the Cisco 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.

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:

Step 1 Choose **Scopes** from the **DHCP** menu to open the List/Add DHCP Scopes page. See Creating Scopes, page 20-10 to create a scope.

Step 2 Click **enabled** for restrict-to-reservations attribute in Miscellaneous Settings group in the Add DHCP Scope 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 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:

Step 1 Choose **Scope Templates** from the **DHCP** menu to open the List DHCP Scope Templates page. See Creating and Applying Scope Templates, page 20-3 to create a scope template.

Step 2 Click **enabled** for restrict-to-reservations attribute in Miscellaneous Settings group in the 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:

Step 1 Choose **Prefixes** from the **DHCP v6** menu to open the List/Add DHCPv6 Prefixes page.

Step 2 Click **Add Prefix** in the List/Add DHCPv6 Prefixes page after entering the prefix name and address.

Step 3 Click the prefix name to open Edit DHCPv6 Prefix page. Click **enabled** for restrict-to-reservations attribute in Non-Parent Settings group.

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

Step 1 To restrict a prefix to client reservations, choose **Prefix Templates** from the **DHCP v6** menu to open the List/Add DHCPv6 Prefix Templates page.

Step 2 Click Add Prefix template button to open the Add DHCPv6 Prefix Template page. Click **enabled** for restrict-to-reservations attribute.

To modify an existing scope 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 the external database assigns the same address to two different clients, 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 Network Registrar 7.1 and later 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 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.

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 **DHCP** menu, choose **Scopes** to open the Manage Scopes (Address Pools) page, then click the View icon (🔍) in the Reservations column to open the List/Add DHCP Reservations for Scope page.

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, choose **Scopes** from the **DHCPv4** menu 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 **DHCP** menu, choose **Reservations** to open the List/Add Reservations page.

Step 2 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**.

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.

Setting Advanced Lease and Reservation Properties

Setting advanced lease and reservation properties can include:

- **Reserving currently leased IP addresses**—See the "Reserving Currently Leased Addresses" section on page 22-16.
- **Unreserving leases**—See the "Unreserving Leases" section on page 22-18.
- **Extending leases to non-MAC addresses**—See the "Extending Reservations to Non-MAC Addresses" section on page 22-18.
- **Forcing lease availability**—See the "Forcing Lease Availability" section on page 22-20.
- **Inhibiting lease renewals**—See the "Inhibiting Lease Renewals" section on page 22-20.
- **Handling leases marked as unavailable**—See the "Handling Leases Marked as Unavailable" section on page 22-21.
- **Setting timeouts for unavailable leases**—See the "Setting Timeouts for Unavailable Leases" section on page 22-22.

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:

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

Step 2 Click the View icon (👓) under the Leases column.

Step 3 Click the IP address of the lease on the List DHCP Leases for Scope page.

Step 4 On the Manage DHCP Lease page, 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 List DHCP Leases for Scope page, the lease will appear as reserved.

Step 6 Modify the scope.

Step 7 To remove the reservation, click **Remove Reservation** on the Manage DHCP Lease 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" section on page 22-20). 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.

Local Advanced Web UI

To unreserve a lease, choose **Reservations** from the **DHCP** menu to open the List/Add Reservations page, then click the Delete icon (🗑) next to 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 Chapter 29, “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 Add DHCP Client-Class page (from the **DHCP** menu, choose **Client-Class**, then **Add Client-Class**) or Edit DHCP Client-Class page (from the **DHCP** menu, choose **Client-Class**, 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 **DHCP** menu, choose **DHCP Server**, then the Local DHCP Server link to open the Edit 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 **DHCP** menu, choose **Policy**, then **Add Policy**) or Edit DHCP Policy page (from the **DHCP** menu, choose **Policies**, 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.

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:

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

Step 2 Click the View icon (🔍) under the Lease column for the scope that has leases.

Step 3 Click the IP address of the lease on the List DHCP Leases for Scope page.

Step 4 On the Manage DHCP Lease page, click **Force Available**. On the List DHCP Leases for 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 leases in the scope to become available.

Inhibiting Lease Renewals

Normally, the Cisco 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 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 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.

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 (click **DHCP**, then **Policies**, 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.

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 the "Pinging Hosts Before Offering Addresses" section on page 22-7.
- **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 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 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 the "Handling Leases Marked as Unavailable" section on page 22-21, 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 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 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.

If a Cisco Network Registrar failover server receives an update from a Cisco Network Registrar DHCP server running prior to Cisco Network Registrar 6.0, the unavailable leases do not have a timeout value. In this case, the upgraded Cisco Network Registrar server uses the *unavailable-timeout* value configured in the scope policy or *system_default_policy* policy as the timeout for the unavailable lease. When the lease times out, the policy causes the lease to transition to available in both failover partners.

Running Address and Lease Reports

You can run these reports on IP addresses and leases:

- **Address Usage**—See the "Running Address Usage Reports" section on page 22-23.
- **Lease History**—See the "Running IP Lease Histories" section on page 22-23.
- **Current Utilization**—See the "Running Lease Utilization Reports" section on page 22-29.
- **Lease Notification**—See the "Receiving Lease Notification" section on page 22-29.

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, in the Leases area, click **List Leases** to open the List DHCP Leases for Scope page. To manage a specific lease, click its IP address on the page. This opens the Manage DHCP Lease 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 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 the "Querying IP Lease History" section on page 22-24.

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 cluster, unless you enable the *ip-history-detail* attribute. 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.

See Also

Enabling Lease History Recording at the Local Cluster
Querying IP Lease History, page 22-24
Trimming Lease History Data, page 22-28

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.

Local Advanced Web UI

To enable lease history recording:

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

Step 2 Click the **Local DHCP** Server link.

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

- ***Lease History*** **(*ip-history*)**—Be sure this is set to v4-only.
- ***ip-history-detail***—Enable this to get detailed lease history data.
- ***ip-history-max-age***—Maximum age of the lease history to collect. With lease history set to v4 only, 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 **Modify Server** at the bottom of the page.

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 enable ip-history**.

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 on page 6-14).

You can adjust the polling criteria for the cluster in the regional cluster web UI by using the attributes described in the "Polling Subnet Utilization and Lease History Data" section on page 6-11.

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

Local and Regional Advanced Web UI

To query the IPv4 lease history, do the following:

Step 1 From the **Address Space v4** menu, choose **Lease History** to open the DHCP v4 Lease History Search page.

You can also go to the DHCP v4 Lease History Search page if you choose **Search** from the **DHCP** menu. If you choose **Search** from the **DHCP** menu, the DHCP v4 Lease Search page is displayed. You have to click the DHCP v4 Lease History Search button to go to the DHCP v4 Lease History Search page.

Note You can use the DHCP v4 Lease Search button in the Local Advanced Web UI to move to DHCP v4 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.

To query the IPv6 lease history, do the following:

Step 1 From the **Address Space v6** menu, choose **Lease History** to open the DHCP v6 Lease History Search page.

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

Note You can use the DHCP v6 Lease 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. 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. The default location is:

- **Windows**—\Program Files\Cisco Network Registrar\bin
- **Solaris and Linux**—/opt/nwreg2/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 Table 22-2.

Table 22-2 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
–4	Output DHCPv4 leases using new interface
–a	Shows the lease attributes, visibility 3.
–b	Displays the local and backup server failover leases.
–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.
–l	Displays output in local time rather than the default GMT.
–m	Displays the local and main server failover leases.
–n *vpn*	Name or ID of an associated VPN, or the word **all** (for all VPNs) or **global** (for IP addresses without a VPN). If omitted, the query is based on the global VPN, or the current one set by the **session set current-vpn** command, unless you use the **all** value with the option.
–o *file*	Sends output to a file.
–v	Displays the database version.
–V *visibility*	Sets the visibility level of the output attributes. The visibility is 3 by default.
–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 or \r\n on Windows). 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**.

Table 22-3 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 22-3 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.

Table 22-3 IP History Query Output Attributes (continued)

Lease Attribute	Description
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-renewal-time	Minimal time that the client is expected to issue a lease renewal.
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 **Servers** menu, choose **Manage Servers** to open the Manage Servers page.

Step 2 Click the **Local CCM Server** link to open the Edit CCM Server page.

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

- ***trim-lease-hist-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.
- ***trim-lease-hist-age***—Provided that the *trim-lease-hist-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 subnet 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 the "Generating Subnet Utilization History Reports" section on page 9-13.

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 or Windows Scheduled 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 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.

See Also

Running Lease Notification Automatically in Solaris and Linux, page 22-30
Running Lease Notification Automatically in Windows, page 22-30
Specifying Configuration Files for Lease Notification, page 22-31

Running Lease Notification Automatically in Solaris and Linux

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/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 the "Specifying Configuration Files for Lease Notification" section on page 22-31).
- 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.

Running Lease Notification Automatically in Windows

Use the Scheduled Tasks service available in Windows Explorer under My Computer to schedule the **lease-notification** command. If you do not find a Scheduled Tasks folder under My Computer, you need to add this optional component from Microsoft Internet Explorer 4.0 or later, or use some third-party task scheduler. You can also use the **at** command to schedule the **nrcmd lease-notification** command. Put multiple entries in the **at** queue, one for each time of day at which you want to run the job.

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 CNR_INSTALL_PATH/conf directory. Cisco 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 Network Registrar strips leading space characters from each line and ignores blank lines.

Querying Leases

Cisco 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 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.

See Also

Leasequery Implementations
Pre-RFC Leasequery for DHCPv4, page 22-32
RFC 4388 Leasequery for DHCPv4, page 22-33
Leasequery for DHCPv6, page 22-34
Leasequery Statistics, page 22-35
Leasequery Example, page 22-36

Leasequery Implementations

Cisco Network Registrar provides three Leasequery implementations:

- **DHCPv4 Cisco-proprietary for pre-RFC 4388**—See the "Pre-RFC Leasequery for DHCPv4" section on page 22-32.
- **DHCPv4 compliant with RFC 4388**—See the "RFC 4388 Leasequery for DHCPv4" section on page 22-33.
- **DHCPv6**—See the "Leasequery for DHCPv6" section on page 22-34.

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 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 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 Network Registrar provides the RFC 4388 implementation alongside the pre-RFC one (see the "Pre-RFC Leasequery for DHCPv4" section on page 22-32) 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 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" section on page 22-42).

The message types for DHCPv6 Leasequery are:

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

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.

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

As of Cisco Network Registrar 7.2, lease queries provide statistics attributes, in the web UI, on the DHCP Server Statistics page (see the "Displaying Statistics" section on page 7-11), 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 Bulk Leasequery.
- **tcp-total-connections**—Number of TCP connections that were opened to the DHCP server for DHCPv6 Bulk Leasequery in this time interval.
- **bulk-leasequeries**—Number of LEASEQUERY packets received over all TCP connections in this time 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.

Leasequery Example

Example 22-2 on page 22-36 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 22-2 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
```

Example 22-3 on page 22-36 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 22-3 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 bulk leasequery

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

Dynamic Lease Notification

The DHCPv4 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 the "DHCP Listener Configuration" section on page 22-42). 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), 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 chapter Chapter 29, “Using Extension Points.”.

Sample Lease Notification Client

Cisco 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.
- An example lnc.properties file. (Run the client with -listprops option for details on the available properties.)
- The Bulk and Active Leasequery Internet Drafts for the Cisco Network Registrar 7.2 implementation.
- A document that details the message values, option codes, and vendor-specific data used for Cisco 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 Network Registrar 7.2.

To extract these items, open the class/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.

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

```
java -cp <classpath>;<installation-path>/examples/dhcp/cnrnotify.jar
com.cisco.cnr.notify.LeaseNotificationClient
```

Note The above command is a one line command, although it is shown as two lines. In the above command, `classpath` should include the location of the lnc.properties file, as well as the log4j and mysql-connector-java-5.1.6 jar files.

Requirements for Sample Java Client

The requirements for the sample Java client are:

- JDK 1.5 or later.
- The java.sql package from JDK 1.5 or later.
- 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 JDBC connector for MySql.

  Note We recommend that you use mysql-connector-java-5.1.6. You may run into problems if you use mysql-connector-java-5.1.7.

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

Local Basic or Advanced 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 **DHCP** menu, choose **DHCP Server** to open the Manage DHCP Server page.

Step 2 Click the Statistics icon() to open the DHCP Server Statistics page.

The Server Statistics details are displayed in this page.

Step 3 Click **Return** to return to the Manage DHCP Server page.

CLI Command

The existing nrcmd **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 **DHCP** menu, choose **Listener**, to open the View/Add DHCP TCP Listener page.

Step 2 Click **Add TCP Listener** to add a new TCP Listener. The Add DHCP TCP Listener page appears.

Step 3 Enter a name in the Name field.

Step 4 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 5 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 v6-server-port for DHCPv6.

Step 6 For Enable attribute, click **true** or **false** radio button. The default value is true.

Step 7 Enter a value for Max-connections, if the default value 10 is not appropriate.

Step 8 Enter a value for Leasequery-backlog-time, if the default value 120 is not appropriate.

Step 9 For leasequery-send-all attribute, click **true** or **false** radio button. The default value is false.

Step 10 Click **Add TCP Listener.**

CLI Commands

The DHCP Listener commands are shown in Table 22-4.

Table 22-4 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** 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>