



Configuring DHCP Scopes and Policies

The Dynamic Host Configuration Protocol (DHCP) is an industry-standard protocol for automatically assigning IP configuration to workstations. DHCP uses a client/server model for address allocation. As administrator, you can configure one or more DHCP servers to provide IP address assignment and other TCP/IP-oriented configuration information to your workstations. DHCP frees you from having to manually assign an IP address to each client. The DHCP protocol is described in RFC 2131. For an introduction to the protocol, see the [“Dynamic Host Configuration and Leases” section on page 2-7](#).

This chapter describes how to set up a DHCP server and its policies. Before clients can use DHCP for address assignment, you must add at least one scope (dynamic address pool) to the server:

- [Chapter 12, “Managing Leases,”](#) describes managing leases in scopes.
- [Chapter 13, “Configuring Clients and Client-Classes,”](#) describes configuring clients and client-classes.
- [Chapter 14, “Managing Advanced DHCP Properties,”](#) describes managing advanced DHCP server properties.
- [Chapter 15, “Configuring Dynamic DNS Update,”](#) explains configuring dynamic DNS update.
- [Chapter 16, “Configuring DHCP Failover,”](#) explains configuring DHCP failover servers.
- [Chapter 17, “Using Extension Points,”](#) explains how to write extensions for special DHCP processing.

Configuring DHCP Servers

When configuring a DHCP server, you must configure the server properties, policies, and associated DHCP options. Network Registrar needs:

- The DHCP server’s IP address.
- One or more scopes—[“Defining and Configuring Scopes” section on page 11-2](#).
- One or more policies, to specify, at a minimum, the lease times for the addresses—See the [“Configuring DHCP Policies” section on page 11-16](#).

General Configuration Guidelines

Here are some guidelines to consider before configuring a DHCP server:

- Separate the DHCP server from secondary DNS servers used for DNS updating—To ensure that the DHCP server is not adversely affected during large zone transfers, it should run on a different cluster than your secondary DNS servers.
- Configure a separate DHCP server to run in remote segments of the wide area network (WAN)—Ensure that the DHCP client can consistently send a packet to the server in under a second. The DHCP protocol dictates that the client receive a response to a DHCPDISCOVER or DHCPREQUEST packet within four seconds of transmission. Many clients, notably early releases of the Microsoft DHCP stack, actually implement a two-second timeout.
- Lease times—See the [“Guidelines for Lease Times”](#) section on page 12-2.

Choosing Server Interfaces

To configure the DHCP server, accept Network Registrar’s defaults or supply the data explicitly:

- Network interface—Ethernet card IP address, which must be static and not assigned by DHCP.
- Subnet mask—Identifies the interface’s network membership. The subnet mask is usually based on the network class of the interface address, in most cases 255.255.255.0.

Network Registrar uses the **default** interface to provide configurable default values for interfaces that the DHCP server discovers automatically.

By default, the DHCP server uses the operating system support to automatically enumerate the active interfaces on the machine and listens on all of them. You can also manually configure the server interface. You should statically configure all the IP addresses assigned to NIC cards on the machine where the DHCP server resides. The machine should not be a BOOTP or DHCP client.

This function is not available in the Web UI.

In the CLI, use the **dhcp-interface** command to manually control which network interface cards’ IP addresses the DHCP server will listen on for DHCP clients. By default, the DHCP server automatically uses all your server’s network interfaces, so use this command to be more specific about which ones to use. See the Usage Guidelines for the **dhcp-interface** command in the *Network Registrar CLI Reference*.

Defining and Configuring Scopes

This section describes how to define and configure scopes for the DHCP server.

Scopes in Network Registrar

A scope consists of one or more ranges of dynamic addresses in a subnet that a DHCP server manages. You must define one or more scopes before the DHCP server can provide leases to clients.

Using DHCP Scope Templates

Scope templates provide a convenient way to define scopes with common properties, rather than having to define these properties for each scope. You can create scope templates on the local and regional cluster.

-
- Step 1** Access is different on the local and regional clusters:
- On the local cluster—On the Primary Navigation bar, click **DHCP**, then **Scope Templates** on the Secondary Navigation bar.
 - On the regional cluster—On the Primary Navigation bar, click **DHCP Configuration**, then **Scope Templates** on the Secondary Navigation bar.
- These actions open the List Scope Templates page for both clusters.
- Step 2** Click **Add Scope Template**. This opens the Add DHCP Scope Template page on both clusters. This page is almost identical to the regional cluster page (see [Figure 4-21 on page 4-31](#)), although the latter has additional push and pull functions, as described in the “[Creating DHCP Scope Templates](#)” section on [page 5-16](#).
- Step 3** Click **Add Scope Template**. You return to the List Scope Templates page, where you can edit or delete the template.
-



Note You must click **Modify Scope Template** to add embedded policies or implement any changes to the scope template properties.

Using Expressions in Scope Templates

You can specify expressions in a scope template to dynamically create scope names, embedded options, and IP address ranges when creating a scope. Expressions can include context variables and operations. These expressions are described in the online help for the Add DHCP Scope Template page.

Creating Scopes

Creating scopes is a local cluster function. Each scope needs to have the following:

- Name
- Policy that defines the lease times, grace period, and options
- Network address and subnet mask
- Range or ranges of addresses

In the local cluster Web UI:

- Step 1** On the Primary Navigation bar, click **DHCP**.
- Step 2** On the Secondary Navigation bar, click **Scopes** to open the List/Add DHCP Scopes page (see [Figure 11-1](#)).

Figure 11-1 List/Add DHCP Scopes Page

- Step 3** Enter a scope name, or leave it blank to use the one defined in the scope name expression of a scope template, if any (see the “Using DHCP Scope Templates” section on page 11-3). In the latter case, choose the scope template. You must always enter a subnet/mask for the scope.
- Step 4** Click **Add Scope**. This opens the Add DHCP Scope page (see Figure 11-2 for the top part of this page).

Figure 11-2 Add DHCP Scope Page

- Step 5** Choose a policy for the scope from the drop-down list. The policy defaults to the *default* policy.
- Step 6** Add ranges for addresses in the scope. The ranges cannot overlap and they must belong to the defined subnet. If you enter just the host number, the range is relative to the netmask.
- Step 7** Click **Add Range** to add each range.

- Step 8** Add any reservations to the scope, which cannot belong to one of the assigned ranges. Add the IP address of the reserved address. Also include its MAC address, in the form **00:d0:ba:d3:bd:3b** or **1,6,00:d0:ba:d3:bd:3b**. Click **Add Reservation** to add each reservation. Define attributes for the scope, if necessary.
- Step 9** Click **Add Scope**.
-

In the CLI:

- To create a scope, use the **scope name create** command. Each scope must identify its network address and mask. When you create the scope, Network Registrar places it in its current virtual private network (VPN), as defined by the **session set current-namespace** command.
- To set the scope's matching policy, use the **scope name set policy** command.
- To add the scope's (space-separated) range of IP addresses, use the **scope name addRange** command.
- To explicitly set the scope's VPN, use the **scope name set namespace-id** command. The VPN must already exist before you can set it for the scope.
- To attach individual options to a scope, use the **scope-policy** command. You might want to do this to define just the *router* option for a particular scope (see the [“Configuring Embedded Policies for Scopes”](#) section on page 11-11).
- Reload the server.

Configuring Multiple Scopes

You can configure multiple scopes (with disjointed address ranges) with the same network number and subnet mask. By default, the DHCP server pools the available leases from all scopes on the same subnet and offers them, in a round-robin fashion, to any client that requests a lease. However, you can also bypass this round-robin allocation by setting an allocation priority for each scope (see the [“Configuring Multiple Scopes Using Allocation Priority”](#) section on page 11-6).

Configuring a single subnet's addresses into multiple scopes helps to organize the addresses in a more natural way for administration. Even though you can configure a virtually unlimited number of leases per scope, if you have a scope with several thousand leases, it can take a while to sort them. This can be a motivation to divide the leases among multiple scopes.

You can divide the leases among the scopes according to the types of leases. Because each scope can have a separate reservations list, you can put the dynamic leases in one scope that has a policy with one set of options and lease times, and all the reservations in another scope with different options and times. Note that in cases where some of the multiple scopes are not connected locally, you should configure the router (having BOOTP relay support) with the appropriate helper address.

Configuring Multiple Scopes for Round-Robin Address Allocation

By default, the DHCP server searches through the multiple scopes in a round-robin fashion. Because of this, you would want to segment the scopes by the kind of DHCP client requests made. When multiple scopes are available on a subnet through the use of secondary scopes, the DHCP server searches through all of them for one that satisfies an incoming DHCP client request. For example, if a subnet has three scopes, only one of which supports dynamic BOOTP, a BOOTP request for which there is no reservation is automatically served by the one supporting dynamic BOOTP.

You can also configure a scope to disallow DHCP requests (the default is to allow them). By using these capabilities together, you can easily configure the addresses on a subnet so that all the DHCP requests are satisfied from one scope (and address range), all reserved BOOTP requests come from a second one, and all dynamic BOOTP requests come from a third. In this way, you can support dynamic BOOTP while minimizing the impact on the address pools that support DHCP clients.

Configuring Multiple Scopes Using Allocation Priority

As of Network Registrar Release 6.1, you can set an allocation priority among scopes instead of the default round-robin behavior described in the previous section. In this way, you can have more control over the allocation process. You can also configure the DHCP server to allocate addresses contiguously from within a subnet and control the blocks of addresses allocated to the backup server when using DHCP server failover (see [Chapter 16, “Configuring DHCP Failover”](#)).

A typical installation would set the allocation priority of every scope by using the *allocation-priority* attribute on the scope. Some installations might also want to enable the *allocate-first-available* attribute on their scopes, although many would not. There is a small performance loss when using *allocate-first-available*, so that you should only use it when absolutely required.

You can control:

- A hierarchy among scopes of which should allocate addresses first.
- Whether to have a scope allocate the first available address rather than the default behavior of the least recently used one.
- Allocating contiguous and targeted addresses in a failover configuration for a scope.
- Priority address allocation server-wide.
- In cases where the scopes have equal allocation priorities set, whether the server should allocate addresses from those with the most or the least number of available addresses.

When there is more than one scope in a network, then the DHCP must decide which scope to allocate an IP address from when it processes a DHCPDISCOVER request from a DHCP client that is not already associated with an existing address. The algorithm that the DHCP server uses to perform this allocation is described in the following section.

Allocation Priority Algorithm

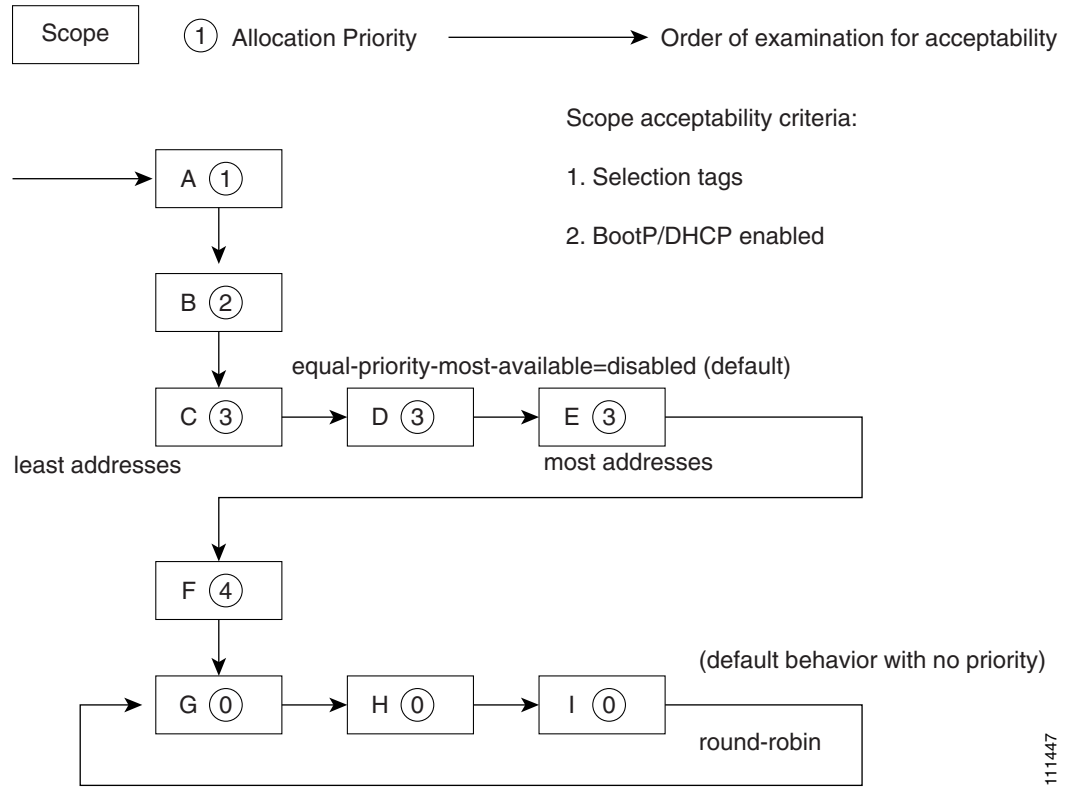
The DHCP server examines the scopes in a Network one at a time to determine if they are acceptable. When it finds an acceptable scope, it tries to allocate an IP address from it to fulfill the DHCPDISCOVER request. The *allocation-priority* scope attribute is used to direct the DHCP server to examine the scopes in a network in a particular order, because in the absence of any allocation priority, the DHCP server examines the scopes in a round-robin order.

[Figure 11-3](#) shows an example of a network with nine scopes (which is unusual, but serves to illustrate several possibilities of using allocation priority). Six of these scopes were configured with an allocation priority, and three of them were not. The server examines the six that were configured with an allocation priority first, in lowest to highest priority order. As the server finds an acceptable scope, it tries to allocate an IP address from it. If the server succeeds, it then finishes processing the DHCPDISCOVER request using this address. If it cannot allocate an address from that scope, it continues examining scopes looking for another acceptable one, and tries to allocate an address from it.

This process is straightforward if no scopes have the same allocation priority configured, but in the case where (as in the example in [Figure 11-3](#)) more than one scope has the same nonzero allocation priority, then the server has to have a way to choose between the scopes of equal priority. The default behavior is to examine the scopes with equal priority starting with the one with the fewest available addresses.

This uses up all of the addresses in one scope before using any others from another scope. This is the situation shown in Figure 11-3. If you enable the *equal-priority-most-available* DHCP server attribute, then the situation is reversed and the scope with the most available addresses is examined first when two scopes have equal priority. This spreads out the utilization of the scopes, and more or less evenly distributes the use of addresses across all of the scopes with equal allocation priority set.

Figure 11-3 Scope Allocation Priority



This *equal-priority-most-available* approach might be used because of another feature in the processing of equal priority scopes. In the situation where there are two scopes of equal priority, if the DHCPDISCOVER request, for which the server is trying to allocate an address, also has a *limitation-id* (that is, it is using the option 82 limitation capability; see the “Subscriber Limitation Using Option 82” section on page 13-10), then the DHCP server tries to allocate its IP address from the same scope as that used by some existing client with the same *limitation-id* (if any). Thus, all clients with the same *limitation-id* tend to get their addresses allocated from the same scope, regardless of the number of available addresses in the scopes of equal priority or the setting of the *equal-priority-most-available* server attribute.

To bring this back to the *equal-priority-most-available* situation, you might configure *equal-priority-most-available* (and have several equal priority scopes), and then the first DHCP client with a particular *limitation-id* would get an address from the scope with the most available addresses (since there are no other clients with that same *limitation-id*). Then all of the subsequent clients with the same *limitation-id* would go into that same scope. The result of this configuration is that the first clients are spread out evenly among the acceptable, equal priority scopes, and the subsequent clients would cluster with the existing ones with the same *limitation-id*.

If there are scopes with and without allocation priority configured in the same network, all of the scopes with a nonzero allocation priority are examined for acceptability first. Then, if none of the scopes were found to be acceptable and also had an available IP address, the remaining scopes without any allocation priority are processed in a round-robin manner. This round-robin examination is started at the next scope beyond the one last examined in this network, except when there is an existing DHCP client with the same *limitation-id* as the current one sending the DHCPDISCOVER. In this case, the round-robin scan starts with the scope from which the existing client's IP address was drawn. This causes subsequent clients with the same *limitation-id* to draw their addresses from the same scope as the first client with that *limitation-id*, if that scope is acceptable and has available IP addresses to allocate.

Address Allocation Attributes

The attributes that correspond to address allocation are described in [Table 11-1](#).

Table 11-1 Address Allocation Priority Settings

Attribute	Type	Description
allocation-priority	Scope (set or unset)	<p>If defined, assigns an ordering to scopes such that address allocation takes place from acceptable scopes with a higher priority until the addresses in all those scopes are exhausted. An allocation priority of 0 (the default) means that the scope has no allocation priority. A priority of 1 is the highest priority, with each increasing number having a lower priority. You can mix scopes with an allocation priority along with those without one. In this case, the scopes with a priority are examined for acceptability before those without a priority.</p> <p>If set, this attribute overrides the DHCP server's <i>priority-address-allocation</i> attribute setting. However, if <i>allocation-priority</i> is unset and <i>priority-address-allocation</i> is enabled, then the allocation priority for the scope is its subnet address. With <i>allocation-priority</i> unset and <i>priority-address-allocation</i> disabled, the scope is examined in the default round-robin fashion.</p>
allocate-first-available	Scope (enable or disable)	<p>If enabled, forces all allocations for new addresses from this scope to be from the first available address. If disabled (the default), uses the least recently used address. If set, this attribute overrides the DHCP server's <i>priority-address-allocation</i> attribute setting. However, if unset and <i>priority-address-allocation</i> is enabled, then the server still allocates the first available address. With <i>allocate-first-available</i> unset and <i>priority-address-allocation</i> disabled, the scope is examined in the default round-robin fashion.</p>

Table 11-1 Address Allocation Priority Settings (continued)

Attribute	Type	Description
failover-backup-allocation-boundary	Scope (set or unset)	<p>If <i>allocate-first-available</i> is enabled and the scope is in a failover configuration, this value is the IP address to use as the point from which to allocate addresses to a backup server. Only addresses below this boundary are allocated to the backup server. If there are no available addresses below this boundary, then the addresses above it are allocated to the backup server. The actual allocation works down from this address, while the normal allocation for DHCP clients works up from the lowest address in the scope.</p> <p>If this attribute is unset or set to zero, then the boundary used is halfway between the first and last addresses in the scope ranges. If there are no available addresses below this boundary, then the first available address is used.</p> <p>See Figure 11-4 on page 11-10 for an illustration of how addresses are allocated in a scope using this setting.</p>
priority-address-allocation	DHCP (enable or disable)	<p>Provides a way to enable priority address allocation for the entire DHCP server without having to configure it on every scope. (However, the scope's <i>allocation-priority</i> setting overrides this one.) If <i>priority-address-allocation</i> is enabled and the scope's <i>allocation-priority</i> attribute is unset, then the scope's subnet address is used for the allocation priority. If the scope's <i>allocate-first-available</i> is unset, then priority address allocation is considered enabled. Of course, when exercising this overall control of the address allocation, the actual priority of each scope depends only on its subnet address, which may or may not be desired.</p>
equal-priority-most-available	DHCP (enable or disable)	<p>By default, when two or more scopes with the same nonzero <i>allocation-priority</i> are encountered, the scope with the least available IP addresses is used to allocate an address for a new client (if that client is not in a limitation list). If <i>equal-priority-most-available</i> is enabled and two or more scopes have the same nonzero allocation priority, then the scope with the most available addresses is used to allocate an address for a new client (if that client is not in a limitation list). In either case, if a client is in a limitation-list, then among those scopes of the same priority, the one that contains other clients in the same list is always used.</p>

Allocating Addresses Within Scopes

When trying to allocate an IP address from within a scope, the default action of the DHCP server is to try to allocate the least recently used address first, although there are a variety of events that can cause an IP address to be used. Thus, in general, there is no way to predict which IP address within a scope is allocated at a given time. Usually this poses no difficulty, but there are times when a more deterministic allocation strategy is desired. To configure a completely deterministic address allocation strategy, you can enable the *allocate-first-available* attribute on a scope. This causes the available address with the lowest numeric value to be allocated for a DHCP client. Thus, the first client gets the first address in the lowest range, and the second client the second one in that range, and so on. This is shown in [Figure 11-4](#).

Figure 11-4 Address Allocation with allocate-first-available Set

Note that there is some minor performance cost to this deterministic allocation strategy, not so much that you should not use it, but possibly enough so that you should not use it if you do not need it. When using this deterministic allocation strategy approach in a situation where the scope is in a failover relationship, the question of how to allocate the available IP addresses for the backup server comes up on the main server. By default, the address halfway between the lowest and highest ones in the scope becomes the *failover-backup-allocation-boundary*. The available addresses for the backup server are allocated working down from this boundary (if any addresses are available in that direction). If no address is available below this boundary, then the first available one above the boundary is used for the backup server. You can configure the *failover-backup-allocation-boundary* for the scope if you wish to have a different address boundary than the halfway point.

You would use a deterministic allocation strategy and configure *allocate-first-available* in situations where you might allocate a scope with a larger number of IP addresses than you were sure you needed. You can later shrink back the ranges in the scope so as to allow moving address space to another network or server. In the nondeterministic approach, the allocated addresses are scattered all over the ranges, and it can be very hard to reconfigure the DHCP clients to free up, say, half of the scope's addresses. However, if you configure *allocate-first-available*, then the allocated addresses tend to cluster low in the scope's ranges. It is then probably simpler to remove ranges from a scope that does not need them, so that those addresses can be used elsewhere.

Editing Scopes

You can edit a scope's properties.

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Modify the fields or attributes on this page as necessary.

- Step 4** To edit the scope's embedded policy, see the [“Configuring Embedded Policies for Scopes”](#) section. To list leases for the scope, see the [“Viewing Leases”](#) section on page 12-1.
- Step 5** Click **Modify Scope**.

In the CLI:

- To look at the properties for all the scopes on the server, use the **scope list** (or **scope listnames**, **scope name show**, or **scope name get attribute** command).
- To reset an attribute, use the **scope name set** command.
- To enable or disable an attribute, use the **scope name enable** or **scope name disable** command.
- To change the subnet and mask of the scope, use the **scope name change-subnet** command.
- To change just the mask, use the **scope name changeMask** command. This changes the *primary-mask* attribute on any secondary scopes, iterates over all reservations and ranges, and displays reservations and ranges that now fall outside the scope.



Note Changing a scope's subnet and mask may result in a warning that certain address ranges have values outside of the new scope definition.

Changing a mask:

- Changes it on the specified scope.
- Changes the *primary-mask* attribute on any secondary scopes to the specified scope.
- Iterates over all reservations in the scope and displays any that now fall outside the scope. If reservations fall outside the scope, then the command returns “101, Ok with warnings” instead of “100 Ok.”
- Iterates over all ranges in the scope and displays any that have endpoints that now fall outside the scope. If range endpoints fall outside the scope, then the command returns “101, Ok with warnings” instead of “100 Ok.”
- If you enable the *delete-orphaned-leases* attribute, at the next DHCP server reload, existing leases are deleted that fall outside the acceptable ranges for this scope and are not in the acceptable ranges of any other scope.

Also, see the [“Making Scopes Secondaries”](#) section on page 11-12.

Configuring Embedded Policies for Scopes

When you create a scope, Network Registrar automatically creates an embedded policy for it. However, the embedded policy has no associated properties or DHCP options until you enable or add them. An embedded policy can be useful, for example, in defining the router for the scope. As the [“Types of Policies”](#) section on page 11-16 describes, the DHCP server looks at the embedded policy of a scope before it looks at its assigned, named policy. The only way to configure an embedded policy is through the Web UI or by using the **scope-policy** command attributes in the CLI.



Tip

In the CLI, the **scope-policy** command uses the same commands as the **policy** command, except that it takes the scope name as an argument.

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Click **Edit Embedded Policy** to open the Edit DHCP Embedded Policy for Scope page.
 - Step 4** Modify the fields, options, and attributes on this page. If necessary, unset attributes.
 - Step 5** Click **Modify Embedded Policy**.
-

In the CLI:

- To determine if there are any embedded property values already set for a scope, use the **scope-policy *scope-name* show** command.
- To enable or disable an attribute, use the **scope-policy *name* enable** or **scope-policy *name* disable** command.
- To set and unset attributes, use the **scope-policy *name* set** and **unset** commands.
- To list, set, and unset vendor options, see the [“Setting Vendor-Specific DHCP Options” section on page 14-5](#)).



Note

If you delete a scope policy, you remove all of its properties and attributes.

Making Scopes Secondaries

Network Registrar supports multiple logical subnets on the same network segment, which are called secondary subnets. With several logical subnets on the same physical network, for example, 192.168.1.0 and 192.168.2.0, you might want to configure DHCP so that it offers addresses from both pools. By pooling addresses this way, you can increase the available number of leases.

To join two logical subnets, create two scopes, and elect one to be primary and the other to be a secondary. After you configure the secondary subnet, any client on this physical network gets a lease from one or the other scope on a round-robin basis, as long as the client does not have a reservation or pre-existing lease.

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#), that you want to make a secondary scope.
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** The first attribute under the Leases area of the page is the *Primary Subnet* attribute. Enter the network address of the subnet of the primary scope, thereby making this a secondary scope.
 - Step 4** Click **Modify Scope**.
-

In the CLI:

- To assign the secondary scope to a primary one, use the **scope *name* set primary-subnet** command, then reload the server.

- To remove the secondary scope, use the **scope name unset primary-subnet** command. When setting the *primary-subnet* attribute, include the number bits for the network mask, using slash notation. For example, represent the network 192.168.1.0 with mask 255.255.255.0 as 192.168.1.0/24. The mask bits are important. If you omit them, a /32 mask (single IP address) is assumed.

It is common practice for the *primary-subnet* to correspond directly to the network address of the primary scope or scopes. For example, with *examplescope1* created in the 192.168.1.0/24 network, associate *examplescope2* with it using *primary-subnet=192.168.1.0/24*. (Note that if Network Registrar finds that the defined subnet has an associated scope, it ignores the mask bit definition and uses the one from the primary scope, just in case they do not match.) However, the *primary-subnet* can be a subnet address that does not have a scope associated with it.

There are three other properties used in previous versions of Network Registrar that denote primary subnet affiliation: *primary-addr*, *primary-mask*, and *primary-scope*. These are present to provide backward compatibility, but should not be used in the current release. The *primary-subnet* attribute (both in the Web UI and CLI) now sets these properties.

Enabling and Disabling BOOTP for Scopes

The BOOTstrap Protocol (BOOTP) was originally created for loading diskless computers. It was later used to allow a host to obtain all the required TCP/IP information so that it could use the Internet. Using BOOTP, a host can broadcast a request on the network and get the data required from a BOOTP server. The BOOTP server listens for incoming requests and generates responses from a configuration database for the BOOTP clients on that network. BOOTP differs from DHCP in that it has no concept of lease or lease expiration. All addresses that a BOOTP server allocates are permanent.

You can configure the Network Registrar DHCP server to act like a BOOTP server. In addition, although BOOTP normally requires static address assignments, you can choose either to reserve addresses (and use static assignments) or have addresses dynamically allocated (known as *dynamic BOOTP*).

When you need to move or decommission a BOOTP client, you can re-use its lease simply by forcing lease availability. See the [“Forcing Lease Availability” section on page 12-9](#).

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Under the BOOTP attributes, enable the *bootp* attribute for BOOTP, or the *dynamic-bootp* attribute for dynamic BOOTP. They are disabled by default.
 - Step 4** Click **Modify Scope**.
-

In the CLI, use the **scope name enable bootp** command to enable BOOTP, and the **scope name enable dynamic-bootp** command to enable dynamic BOOTP. Reload the DHCP server.

Disabling DHCP for Scopes

You can disable DHCP for a scope if you want to use it solely for BOOTP. See the [“Enabling and Disabling BOOTP for Scopes” section](#). You can also temporarily de-activate a scope by disabling DHCP, but it is more often used if you are enabling BOOTP. See the [“De-activating Scopes” section on page 11-14](#).

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Under the BOOTP attributes, disable the *dhcp* attribute and enable the *bootp* attribute.
 - Step 4** Click **Modify Scope**.
-

In the CLI, use the **scope name disable dhcp** command to disable DHCP. You should also enable BOOTP and reload the server.

De-activating Scopes

You may want to temporarily de-activate all the leases in a scope. To do this, you must disable both BOOTP and DHCP for the scope.

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Under the Miscellaneous attributes, explicitly enable the *deactivated* attribute.
 - Step 4** Click **Modify Scope**.
-

In the CLI, use the **scope name enable deactivated** command to disable BOOTP and DHCP for the scope. Reload the DHCP server.

Setting Scopes to Renew-Only

You can control whether to allow existing clients to re-acquire their leases, but not to offer any leases to new clients. A renew-only scope does not change the client associated with any of its leases, other than to allow a client currently using an available IP address to continue to use it.

In the local cluster Web UI:

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 11-3](#).
 - Step 2** Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.
 - Step 3** Under the Miscellaneous attributes, explicitly enable the *renew-only* attribute.
 - Step 4** Click **Modify Scope**.
-

In the CLI, use the **scope name enable renew-only** command to set a scope to renew-only.

Setting Free Address SNMP Traps on Scopes

You can set SNMP traps to capture unexpected free address events by enabling the traps and setting the low and high thresholds for a scope. These scope settings override any settings you make on the server level using the **trap** command in the CLI, which are:

```
nrcmd> trap enable free-address-low
nrcmd> trap enable free-address-high
nrcmd> trap set free-address-low-threshold=20%
nrcmd> trap set free-address-high-threshold=25%
```

To override these settings for each scope, when you create or edit the scope, look at the attributes in the SNMP Trap Settings area of the Add or Edit Scope page, or the attributes for the **dhcp** CLI command:

- *trap-free-address-low*
- *trap-free-address-high*
- *trap-free-address-low-threshold*
- *trap-free-address-high-threshold*

The first two attributes are enable/disable settings. The high setting is enabled by default, although it is less significant than, and meant only to re-arm, the low address setting. The last two attributes are the corresponding threshold settings for each, expressed as percentages of free space. If there is no threshold value specified, it adopts the server threshold value as set by the **trap** command. When setting the threshold values, it is advisable to maintain a small offset between the low and high values, as described in the “SNMP Troubleshooting” section on page 2-22), even though both values default to 20% globally. The offset can be as little as 5%, for example, a low value of 20% and a high value of 25%.

Here are some variations on how you can set the server and scope values for these attributes:

- Get each scope to trap and reset the free address values based on the server settings, as long as at least one recipient is configured.
- Disable the traps at the scope level or specify different percentages for each scope.
- Disable the traps globally on the server, but turn them on for different scopes.

Removing Scopes



Caution


Although removing a scope from a DHCP server is easy to do, be careful. Doing so compromises the integrity of your network. There are several ways to remove a scope from a server, either by re-using or not re-using addresses, as described in the following sections.

DHCP, as defined by the IETF, provides an address lease to a client for a specific time (defined by the server’s administrator). Until that time elapses, the client is free to use its leased address. A server cannot revoke a lease and stop a client from using an address. Thus, while you can easily remove a scope from a DHCP server, the clients that obtained leases from it can continue to do so until it expires. This is true even if the server does not respond to their renewal attempts, which happens if the scope was removed.

This does not present a problem if the addresses you remove are not re-used in some way. However, if the addresses are configured for another server before the last lease expires, the same address might be used by two clients, which can destabilize the network.

Removing Scopes if Not Re-using Addresses

If you do not plan to re-use the addresses from the scope, you can remove the scope from the server.

In the local cluster Web UI, if you are sure you do not plan to re-use the scope, on the List/Add DHCP Scope page, click the Delete icon () next to its name, and confirm or cancel the deletion.

In the CLI, be sure you are not immediately planning to re-use the addresses in the scope, then use the `scope name delete` command to delete it.

Removing Scopes if Re-using Addresses

If you want to re-use the addresses after removing a scope, you have two options:

- If you can afford to wait until all the leases in the scope expire—Remove the scope from the server, then wait for the longest lease time set in the policy for the scope to expire. This ensures that no clients are using any addresses from that scope. You can then safely re-use the addresses.
- If you cannot afford to wait until all the leases in the scope expire—Do not remove the scope. Instead, de-activate the scope. See the “[De-activating Scopes](#)” section on page 11-14. Unlike a scope that was removed, this causes the server to refuse all clients’ renewal requests, which forces many of them to request a new lease. This moves these clients more quickly off the de-activated lease than if the scope had been removed.

You can also use the `ipconfig` utility in Windows to cause a client to release (`/release`) and re-acquire (`/renew`) its leases, thereby moving it off a de-activated lease immediately. You can only issue this utility from the client machine, which makes it impractical for a scope with thousands of leases in use. However, it can be useful in moving the last few clients in a Windows environment off de-activated leases in a scope.

Configuring DHCP Policies

Every DHCP server must have one or more *policies* defined for it. Policies define lease duration, gateway routers, and other configuration parameters, in what are called DHCP options. Policies are especially useful if you have multiple scopes, because you need only define a policy once and apply it to the multiple scopes.

You can define named policies with specific option definitions or you can use system defaults. This section describes how to configure a policy both ways.

Types of Policies

There are three types of policies—system default, named, and embedded policies:

- System default (**system_default_policy**)—Provides a single location for setting default values on certain options for all scopes. Use the system default policy to define standard DHCP options that have common values for all clients on all the networks that the DHCP server supports. You can modify the system default options and their values. If you delete a system default policy, it re-appears using its original list of options and their system-defined values (see [Table 11-2](#)). These options are visible when using the `policy name listOptions` command in the CLI, and in the GUI on the Policies tab of the DHCP server properties dialog box.

Table 11-2 System Default Policy Option Values

System Default Option	Predefined Value
all-subnets-local	False
arp-cache-timeout	60 seconds
broadcast-address	255.255.255.255
default-ip-ttl	64
default-tcp-ttl	64
dhcp-lease-time	604800 seconds (7d)
ieee802.3-encapsulation	False
interface-mtu	576 bytes
mask-supplier	False
max-dgram-reassembly	576 bytes
non-local-source-routing	False
path-mtu-aging-timeout	6000 seconds
path-mtu-plateau-tables	68, 296, 508, 1006, 1492, 2002, 4352, 8166, 17914, 32000
perform-mask-discovery	False
router-discovery	True
router-solicitation-address	224.0.0.2
tcp-keepalive-garbage	False
tcp-keepalive-interval	0 seconds
trailer-encapsulation	False

- **Named**—Policies you explicitly define by name. Named policies are usually named after their associated scope or client grouping. For example, they might be assigned options that are unique to a subnet, such as for its routers, and then be assigned to the appropriate scope.



Tip Network Registrar includes a **default** policy when you install the DHCP server. The server assigns this policy to newly created scopes. You can modify the default policy or assign another one to the scope.

- **Embedded**—A policy embedded in (and limited to) a named scope, client, or client-class. An embedded policy is implicitly created (or removed) when you add (or remove) the corresponding object, such as a scope. However, the embedded policy options have no default values and are initially undefined. See the [“Configuring Embedded Policies for Scopes”](#) section on page 11-11.

Policy Reply Options

To eliminate any conflicting option values that are set at these various levels, the Network Registrar DHCP server uses a local priority method. It adopts the more locally defined option values first, ignores the ones defined on a more global level, and includes any default ones not otherwise defined. Before returning option values to a DHCP client, the server prioritizes the option values in this order:

1. Client embedded policy.
2. Client assigned policy.
3. Client-class embedded policy.
4. Client-class assigned policy.
5. Scope embedded policy for clients, or address block embedded policy for subnets.
6. Scope assigned policy for clients (or default policy if a named policy is not applied to the scope), or address block assigned policy for subnets.
7. Else any remaining unfulfilled options are checked in the `system_default_policy`.

Then, the server looks through the policies for a reply-options list. It looks for `bootp-` or `dhcp-reply-options`, depending on the client. The server uses the first list it finds. For each option in the list, the server looks through all of the policies, in order, and returns the data from the first policy that has a matching option.

Creating Policies

This section describes how to create a policy at the DHCP server level and then allow a specific scope or scopes to reference it. A policy can consist of a:

- Name—Must be unique, even if the character case is different.
- Permanent lease option—A permanent lease never expires.
- Lease time—How long a client can use an assigned lease before having to renew the lease with the DHCP server (not available for an embedded policy). The default lease time for both system default and default policies is seven days (604800 seconds).

A policy contains two lease times—the client lease time and the server lease time:

- Client lease time—Set through the `setLeaseTime` keyword. This lease time determines how long the client believes its lease is valid.
- Server lease time—Set through the `server-lease-time` attribute. This lease time determines how long the server considers the lease valid. Note that the server lease time is independent of the lease's grace period. The server does not allocate the lease to another client until after the lease time and grace period expire.



Caution

Although Network Registrar supports the use of two lease times for special situations, Cisco Systems generally recommends that you not use the `server-lease-time` attribute.

You can establish these two different lease times if you want to retain information about clients' DNS names and yet have them renew their leases frequently. When you use a single lease time and it expires, the server no longer keeps that client's DNS name. However, if you use a short client lease time and a longer server lease time, then the client information is retained even after the client's lease expires.

- Lease grace period—Time period after the lease expires that it is unavailable for re-assignment (not available for an embedded policy).
- DHCP options and their defined values—See [Appendix B, “DHCP Options.”](#)

In the local cluster Web UI:

- Step 1** On the Primary Navigation bar, click **DHCP**.
- Step 2** On the Secondary Navigation bar, click **Policies** to open the List DHCP Policies page (see [Figure 11-5](#)).

Figure 11-5 List DHCP Policies Page

Name	Offer Timeout	Grace Period	DHCP Lease Time (option 51)
default	2m	5m	1w
system_default_policy	2m	5m	1w

Navigation: Home | Administration | Zone | Host | Address Space | **DHCP**
 Scopes | Scope Templates | VPNs | **Policies** | Clients | Client-Classes | Networks | Failover | DHCP Server

Buttons: Add Policy

Search: [Name]

Page: 10 Change Page Size

- Step 3** There will be two policies initially listed—default and system_default_policy. To add an additional policy, click **Add Policy** to open the Add DHCP Policy page.
- Step 4** Give the policy a unique name.
- Step 5** Either accept the offer timeout and grace period defaults or set them differently.
- Step 6** See the [“Adding DHCP Options for Policies” section on page 11-20](#) for adding options for the policy.
- Step 7** Click **Add Policy** to add the policy.

In the CLI:

- Use the **policy name create** command to create the policy.
- Use the **policy set attribute** command to set the lease options (in this example, the lease grace period). Policy names are not case-sensitive.
- To set permanent leases for the policy, use the **policy name enable permanent-leases** command.
- To set the policy’s domain name, name server, and routers, use the **policy name setOption** command.
- To set the policy’s lease time, use the **policy name setLeaseTime** command.
- To confirm, use the **policy name listOptions** or **policy name getOption dhcp-lease-time** command.
- To set the subnet mask, you have to use a combination of the **policy name setOption subnet-mask** command and the **dhcp enable get-subnet-mask-from-policy** command.
- To remove the subnet mask from the policy, either unset the attribute or disable it.
- Reload the DHCP server.

Adding DHCP Options for Policies

DHCP options supply configuration parameters automatically to DHCP clients, such as their domain, and their name server and subnet router addresses. See [Appendix B, “DHCP Options.”](#)

You can view, set, unset, and edit individual option values. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed for the given option name. Network Registrar DHCP options are grouped into categories to aid you in identifying options that you must set in various usage contexts. [Table B-11 on page B-18](#) describes the categories. You can also create custom options. The [“Configuring Custom DHCP Options” section on page 14-10](#) describes these options.

In the local cluster Web UI:

-
- Step 1** Create a policy, as described in the [“Creating Policies” section on page 11-18](#).
 - Step 2** Add DHCP options to the policy by clicking their numbers and names in the Number drop-down list. The choices indicate the datatype of the option value.
 - Step 3** Add the appropriate option value in the Value field. The Web UI does error checking based on the value entered. For example, to add the lease time for the policy, click the *[51] dhcp-lease-time (unsigned time)* option in the Number drop-down list, then add a lease time value in the Value field.
 - Step 4** Click **Add Option** for each option. You must supply a value or you cannot add the option.
 - Step 5** Click **Add Policy** to add the policy.
-

In the CLI:

- To view option values, use the **policy name** **getOption** and **policy name** **listOptions** commands.
- To set option values, use the **policy name** **setOption** *option* command. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed, for the given option name. For a list of the options, use the **dhcp-option list** command.
- To unset option values, use the **policy name** **unsetOption** command.

Editing Embedded Policies

You can edit the embedded policy for a scope, scope template, client, and client-class. An embedded policy is implicitly created when you create one of these objects. You need to specify an offer timeout, grace period, and server lease time value for the embedded policy. You can also add DHCP options and further attributes for the embedded policy.

In the local cluster Web UI:

-
- Step 1** On the Primary Navigation bar, click the **DHCP** tab.
 - Step 2** On the Secondary Navigation bar, click the **Scopes, Scope Templates, Clients, or Client-Classes** tab.
 - Step 3** Click the name of a scope, template, client, or client-class to open the Edit page for that object.
 - Step 4** Click **Edit Embedded Policy** under the Embedded Policy section of the page. This opens the Edit DHCP Embedded Policy page for the object (see [Figure 11-6](#) for a client-class embedded policy).

Figure 11-6 Edit DHCP Embedded Policy Page

Attribute	Value	Unset?
Offer timeout	2m	<input type="checkbox"/>
Grace period	5m	<input type="checkbox"/>

Options	Number	Value

Add Option

Attribute	Value	Data Type	Default	Unset?
boot-reply-options		list		<input type="checkbox"/>
dhcp-reply-options		list		<input type="checkbox"/>

Step 5 Click one of the **Modify** buttons.



Note You must click **Modify...** on the next page that comes up to implement the embedded policy changes.

Managing DHCP Networks

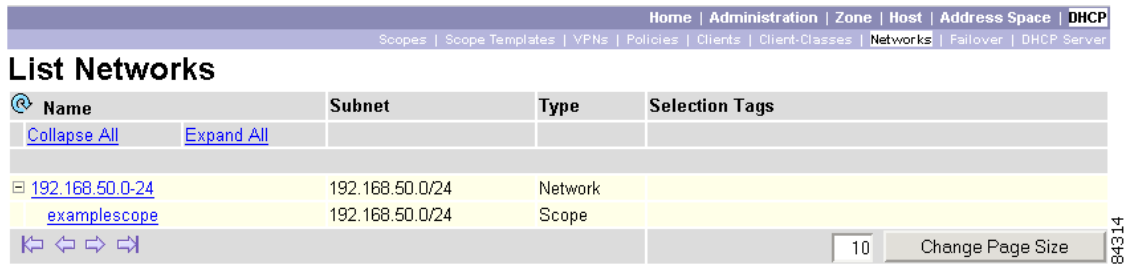
When you create a scope in the Web UI, this creates a network based on the subnet and mask you specify for the scope. Scopes can share the same subnet, so it is often convenient to show the networks and the scopes associated with them. Managing these networks is a local cluster function only. You can also edit the name of any created network.

Listing Networks

The List Networks page lets you list the networks created by scopes and determine to which scopes the networks relate. The networks are listed by name, which the Web UI creates from the subnet and mask. On this page, you can expand and collapse the networks to show or hide their associated scopes.

In the local cluster Web UI, on the Primary Navigation bar, click **DHCP**. On the Secondary Navigation bar, click **Networks**. This opens the List Networks page (see [Figure 11-7](#)).

Figure 11-7 List Networks Page



On the List Networks page, you can:

- List the networks—The networks appear alphabetically by name and identify their subnet and any assigned scope-selection tags. Click the + sign next to a network to expand the view to show the associated scopes. To expand all the network views, click **Expand All**; to collapse all the network views to show just the network names, click **Collapse All**.
- Edit a network name—Click the network name. See the “Editing Networks” section.

Editing Networks

You can edit a network name. The original name is based on the subnet and mask as specified in the scope. You can change this name to an arbitrary but descriptive string. In the local cluster Web UI:

-
- Step 1** On the Primary Navigation bar, click **DHCP**.
 - Step 2** On the Secondary Navigation bar, click **Network**. This opens the List Networks page.
 - Step 3** Click the name of the network you want to edit. This opens the Edit Network page.
 - Step 4** Edit the network data.
 - Step 5** Click **Modify Network**.
-