



Configuring Scopes and Networks

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 [Chapter 18, “Introduction to Dynamic Host Configuration.”](#)

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

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 (see the [“Defining and Configuring Scopes”](#) section on page 19-2).

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 21-2.


Configuring DHCP 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.

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.

You can configure the network interfaces for the DHCP server from the Manage Servers page in the Web UI at the local cluster. To get there, click **Servers**, then **Manage Servers**. Click the Interfaces icon  for the DHCP server to open the Manage DHCP Server Network Interfaces page.

This page shows the available network interfaces that you can configure for the server. By default, the server uses all of them. To configure an interface, click the Edit icon  in the Configure column for the interface. This adds the interface to the Configured Interfaces table, where you can edit or delete it. Clicking the name of the configured interface opens the Edit DHCP Server Network Interface page, where you can change the address and port of the interface. Click **Modify Interface** when you are done editing, then click **Return** to return to the Manage Servers page.

In the CLI, use **dhcp-interface** 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. 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.

Creating and Applying Scope Templates

Scope templates apply certain common attributes to multiple scopes. These common attributes include a scope name based on an expression, policies, address ranges, and an embedded policy option based on an expression. The scope templates you add or pull from the local clusters are visible on the List DHCP Scope Templates page.

You get to this page by clicking **DHCP**, then **Scope Templates**. This functionality is available only to administrators assigned the dhcp-management subrole of the central-cfg-admin role at the regional cluster and ccm-admin at the local cluster.

To explicitly create a scope template, click **Add Scope Template** on this page. This opens the Add DHCP Scope Template page, which includes a number of fields and settings (see [Figure 4-19 on page 4-31](#)). You must give the template at least a name. You can also choose an existing policy for the scope template. The other fields require expression settings that are described in the following sections.

In the CLI, create a scope template using **scope-template name create**. For example:

```
nrcmd> scope-template example-scope-template create
```

You can also associate a policy with the scope template:

```
nrcmd> scope-template example-scope-template set policy=examplepolicy
```

The attributes involving expressions are described in the following sections.

Using Expressions in Scope Templates

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



Note

Expressions are not the same as DHCP extensions. Expressions are commonly used to create client identities or look up clients. Extensions (see [Chapter 28, “Using Extension Points”](#)) are used to modify request or response packets.

There are three fields on the Add DHCP Scope Template page for which you must specify an expression:

- Scope name—Must return a string.
- Address range—Must return IP addresses.
- Embedded policy options.

The CLI does not have a scope name expression equivalent, but it does have attributes that set the address range and embedded policy option using expressions:

- *ranges-exp*
- *options-exp*



Note

If you apply the template to a scope that already has ranges defined, the address range expression of the scope template is not evaluated for that scope.

Scope Name Expression Example

You might want to set an expression so that the template constructs scope names starting with “ISP–” and followed by the subnet of the scope and a derivative of its ping timeout value. You would use the following expression in the Scope Name Expression field:

```
(concat "ISP-" subnet "-" (+ template.ping-timeout 10))
```

The elements of the example expression are:

- **(concat ...)**—Concatenation operation, which concatenates all the following values into one value.
- **“ISP–”**—String with which to start the scope name.
- **subnet**—Keyword variable that indicates to use the existing subnet defined for the scope.
- **“–”**—Indicates to include this hyphen to construct the value.
- **(+ template.ping-timeout 10)**—Indicates to add the *ping-timeout* property value for the scope to the number 10.

If the scope’s subnet happens to be 192.168.50.0/24 and its *ping-timeout* value 100, the resulting constructed scope name would be:

```
ISP-192.168.50.0/24-110
```

Range Expression Example

You might want to set an expression so that the template constructs only certain address ranges for scopes. You can either be explicit about the actual starting and ending addresses, or you can make them relative to the subnet. Here are two ways of requesting relative ranges in the Range Expression field:

```
(create-range first-addr last-addr)
(create-range 1 10)
```

The first **create-range** operation creates the address range based on the first through last usable address in the subnet. For the 192.168.50.0/24 subnet, for example, the address range would be 192.168.50.1 through 192.168.50.254. Because the second operation specifies integers instead of full IP addresses, it makes the range relative to the subnet based on its mask. If the template discovers the subnet to be 192.168.50.0/26, it takes the first through tenth address in this subnet, which would be 192.168.50.65 through 192.168.50.74.

In the CLI, you would set the range expression with the second operation as follows:

```
nrcmd> scope-template example-scope-template set ranges-expr=(create-range 1 10)
```

Embedded Policy Option Expression Example

An embedded policy is important because the DHCP server looks at it before it looks at the scope's assigned, named policy. This is usually where you would set the DHCP options on a scope. You might want to set an expression so that the template constructs DHCP options for the scope's embedded policy. Here are some examples:

```
(create-option "domain-name" "example.com")
(create-option 3 "10.10.10.1")
(create-option "routers" (create-ipaddr subnet 10))
```

The first **create-option** operation associates the value `example.com` with the *domain-name* option for the scope. The second operation associates the address 10.10.10.1 with the *routers* option (number 3). The third operation creates an IP address for the *routers* option based on the tenth address in a subnet.

In the CLI, you would set the policy option expression with the first operation as follows:

```
nrcmd> scope-template example-scope-template set options-expr=(create-option domain-name
example.com)
```

Additional Scope Template Attributes

The optional additional attributes appear in functional categories. For a description of each attribute, click the attribute name to open a help window. For example, you might want to enable dynamic DNS updates for the scope, or set the main and backup DHCP failover servers.

After you complete these fields, click **Add Scope Template**.

Editing Scope Templates

To edit a scope template, click its name on the List DHCP Scope Templates page. The Edit DHCP Scope Template page is essentially the same as the Add DHCP Scope Template page, except for an additional attribute unset function. Make your changes, then click **Modify Scope Template**.

In the CLI, edit a scope template's attributes using **scope-template name set attribute**. For example:

```
nrcmd> scope-template example-scope-template set policy=default
```

Applying Scope Templates to Scopes

You can apply a scope template to a scope in a few ways:

- In the Web UI with a named scope—On the List/Add DHCP Scopes page, include the name of the scope, add its subnet/mask, then choose the scope template from the drop-down list. This creates a scope with the name specified and with the attributes set for the scope template, including the expressions you might have set.
- In the Web UI with a derived scope name—If you set a scope name expression for the scope template, on the List/Add DHCP Scopes page, omit the name of the scope, but add its subnet/mask, then choose the scope template from the drop-down list. This creates a scope with a name synthesized from the scope name expression. If you do not set a scope name expression in the template and apply it to the scope without specifying a name for the scope, you get an error.
- In the CLI when creating a scope—Create the scope using **scope name create address mask template=template-name**. For example:

```
nr cmd> scope example-scope create 192.168.50.0 24 template=example-scope-template
```

- In the CLI with existing scopes—Use **scope-template name apply-to {all | scope1,scope2,...}**. For example:

```
nr cmd> scope-template example-scope-template apply-to examplescope-1,examplescope-2
```



Caution

Be careful applying a scope template to existing scopes, and especially using the **apply-to all** method. The template overwrites all the scopes' attributes with its own, which can have a detrimental effect if the scopes are active.

Cloning a Scope Template

In the CLI, you can also clone a scope template from an existing one by using **scope-template clone-name create clone=template**, and then make adjustments to the clone. For example:

```
nr cmd> scope-template cloned-template create clone=example-scope-template-1
ping-timeout=200
```

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

Step 1 In the local Web UI, click **DHCP**, then **Scopes** to open the List/Add DHCP Scopes page (see [Figure 19-1](#)).

Figure 19-1 List/Add DHCP Scopes Page (Local)

- Step 2** 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 “[Creating and Applying Scope Templates](#)” section on page 19-2). In the latter case, choose the scope template. You must always enter a subnet/mask for the scope.

In the CLI, use **scope name create**. 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 **session set current-vpn**.

In the CLI, to explicitly set the scope’s VPN, use **scope name set vpn-id**. The VPN must already exist before you can set it for the scope.

- Step 3** In the Web UI, click **Add Scope**. This opens the Add DHCP Scope page (see [Figure 19-2](#) for a partial view of this page).

Figure 19-2 Add DHCP Scope Page (Local)

- Step 4** In the Web UI, choose a policy for the scope from the drop-down list. The policy defaults to the *default* policy.

In the CLI, use **scope name set policy**.

- Step 5** Add ranges for addresses in the scope. The ranges can be any subset of the defined scope, but cannot overlap. If you enter just the host number, the range is relative to the netmask. Do not enter ranges that include the local host or broadcast addresses (usually 0 and 255).
- In the Web UI, add the range, then click **Add Range** to add each range.
- In the CLI, use `scope name addRange`.
- Step 6** Add any reservations to the scope. 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. A reservation can alternatively consist of a lookup key of a MAC address, string, or binary value.
- Step 7** Click **Add Scope**.
-

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 19-8).

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 26, “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.

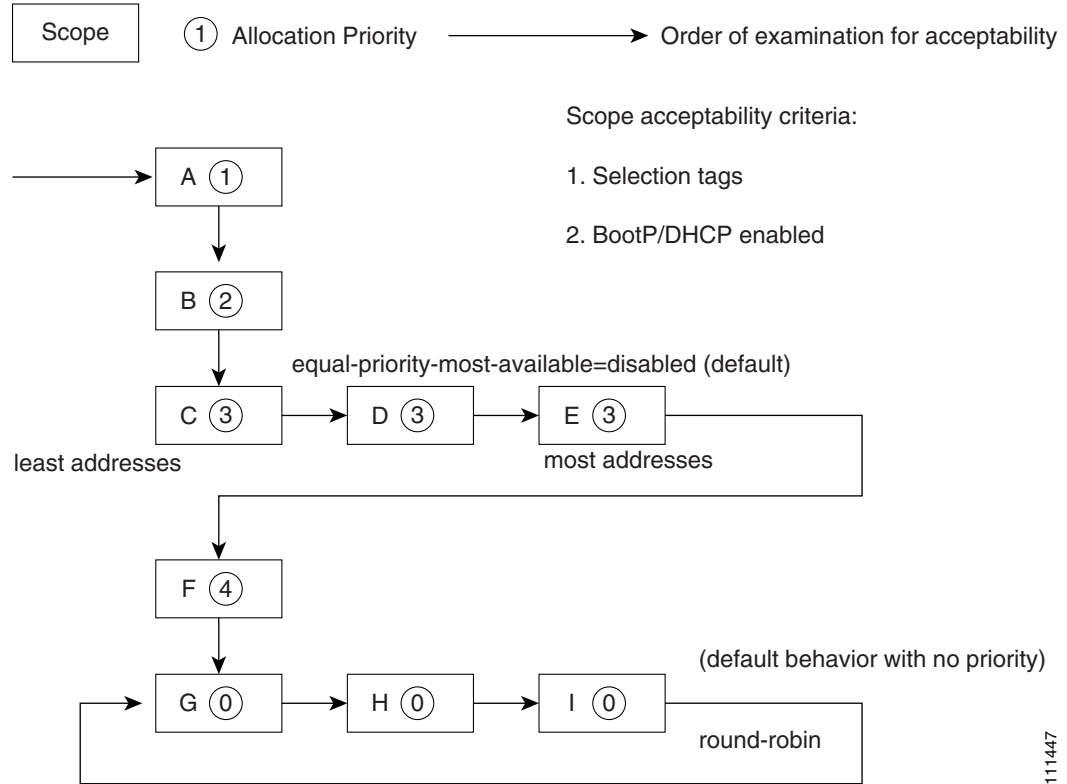
When there is more than one scope in a network, then the DHCP must decide which scope to allocate an IP address from when it processes a DHCPDISCOVER request from a DHCP client that is not already associated with an existing address. The algorithm that the DHCP server uses to perform this allocation 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 19-3](#) shows an example of a network with nine scopes (which is unusual, but serves to illustrate several possibilities of using allocation priority).

Figure 19-3 Scope 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 19-3) 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 19-3. 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.

You can use this *equal-priority-most-available* approach 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 23-12), 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 19-1](#).

Table 19-1 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 19-1 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 19-4 on page 19-12 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 19-4](#).

Figure 19-4 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



Tip

The Web UI indicates the current synchronization status of the scope on the Edit DHCP Scope page. The CLI equivalent is **scope list**, which shows the current synchronization status.

Step 1 Create a scope, as described in the [“Creating Scopes” section on page 19-5](#).

Step 2 In the local Web UI, click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page.

In the CLI, to look at the properties for all the scopes on the server, use **scope list** (or **scope listnames**, **scope name show**, or **scope name get attribute**).

Step 3 Modify the fields or attributes as necessary.

In the CLI:

- To reset an attribute, use **scope name set**.
- To enable or disable an attribute, use **scope name enable** or **scope name disable**.
- To change the subnet and mask of the scope, use **scope name change-subnet**.
- To change just the mask, use **scope name changeMask**. 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`.
- While enabling the *delete-orphaned-leases* attribute, at the next DHCP server reload, deletes existing leases that fall outside of the acceptable ranges for this scope and are not in the acceptable ranges of any other scope.

Step 4 To edit the scope’s embedded policy, see the [“Configuring Embedded Policies for Scopes” section on page 19-14](#). To list leases for the scope, see the [“Viewing Leases” section on page 21-1](#).

Step 5 Click **Modify Scope**.

Staged and Synchronous Mode

New scopes or modifications to scopes can be in one of two modes—staged or synchronous:

- **Staged**—New scopes or modifications to existing scopes are written to the database, but not propagated to the DHCP server until the DHCP server is reloaded.
- **Synchronous**—Most new scopes and scope modifications are immediately propagated to the DHCP server (without the need for a reload). Not all scope changes are possible. For example, changing the primary subnet on a scope is not allowed (a reload is required to effect the change). Furthermore, only scope attribute changes can be propagated without a reload. For example, changes to named policies require a DHCP server reload. If a scope was deleted (which is never synchronous) and then a new scope with the same name is added, a DHCP server reload is required to load that new scope.

If a scope is added or modified while in staged mode and then the scope edit mode is changed to synchronous, the first change in synchronous mode applies all pending changes for that scope (not just the ones made while in synchronous mode).

To view the current scope edit mode or change the scope edit mode, go to the Home page under Session Settings. When editing a scope, if the scope is up to date in the DHCP server, the “Scope *name* status: synchronized” status is displayed on the Edit DHCP Scope page. If the scope is not up to date, the “Scope *name* status: reload required” message is displayed.

In the CLI, you can view the scope edit mode by using **session get scope-edit-mode**, or set the scope edit mode using **session set scope-edit-mode={sync | staged}**. To view the scopes that are not synchronized with the DHCP server, use **scope report-staged-edits**. For example:

```
nrcmd> scope report-staged-edits
100 Ok
example-scope: [reload-required]
```

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 20-1 describes, the DHCP server looks at the embedded policy of a scope before it looks at its assigned, named policy.

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

-
- Step 1** Create a scope, as described in the “[Creating Scopes](#)” section on page 19-5.
 - Step 2** In the local Web UI, 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 **scope-policy scope-name show**.
- To enable or disable an attribute, use **scope-policy name enable** or **scope-policy name disable**.

- To set and unset attributes, use **scope-policy name set** and **unset**.
- To list, set, and unset vendor options, see the [“Setting Custom and Vendor-Specific DHCP Option Definitions” section on page 20-8](#)).

**Note**

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

Configuring Multiple Subnets on a Network

Network Registrar supports multiple logical subnets on the same network segment, which are also called secondary subnets. With several logical subnets on the same physical network, for example, 192.168.1.0/24 and 192.168.2.0/24, 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, a new client on this physical network gets a lease from one or the other scope on a round-robin basis.

-
- Step 1** In the local cluster Web UI, create a scope (see the [“Creating Scopes” section on page 19-5](#)) that you will make a secondary scope.
- Step 2** In the local Web UI, 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.

In the CLI:

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

Three other properties used in previous versions of Network Registrar denote primary subnet affiliation: *primary-addr*, *primary-mask*, and *primary-scope*. These properties 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.

- Step 4** In the Web UI, click **Modify Scope**.
- Step 5** Restart or reload the server.
-

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 21-11](#).

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 19-5](#).
 - Step 2** In the local cluster Web UI, 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.

In the CLI, use **scope name enable bootp** to enable BOOTP, and **scope name enable dynamic-bootp** to enable dynamic BOOTP. Reload the DHCP server.
 - Step 4** In the Web UI, click **Modify Scope**.
-

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 on page 19-16](#). You can also temporarily deactivate a scope by disabling DHCP, but deactivation is more often used if you are enabling BOOTP. See the [“Deactivating Scopes” section on page 19-17](#).

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 19-5](#).
 - Step 2** In the local cluster Web UI, 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.

In the CLI, use **scope name disable dhcp** to disable DHCP. You should also enable BOOTP and reload the server.
 - Step 4** In the Web UI, click **Modify Scope**.
-

Deactivating Scopes

You might want to temporarily deactivate all the leases in a scope. To do this, you must disable both BOOTP and DHCP for the scope.

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 19-5](#).
 - Step 2** In the local cluster Web UI, 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.
In the CLI, use **scope name enable deactivated** to disable BOOTP and DHCP for the scope. Reload the DHCP server.
 - Step 4** In the Web UI, click **Modify Scope**.
-

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.

-
- Step 1** Create a scope, as described in the [“Creating Scopes” section on page 19-5](#).
 - Step 2** In the local cluster Web UI, 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.
In the CLI, use **scope name enable renew-only** to set a scope to renew-only.
 - Step 4** In the Web UI, click **Modify Scope**.
-

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. You can also set traps based on networks and selection tags instead of scopes.

-
- Step 1** In the Web UI, create a trap configuration by clicking DHCP, then Traps to open the List Trap Configurations page.
 - Step 2** Click **Add Trap Configuration** to open the Add Trap Configuration page (see [Figure 1-2 on page 1-4](#)).
 - Step 3** Enter a name for the trap configuration, choose **scope** from the mode drop-down list, and enter the low and high threshold values (they are 20% and 25%, respectively, by default). Click **Add Trap Configuration**. (You can go back to edit these values if you need to.)

In the CLI, use **addr-trap name create** to add a trap configuration. To set the thresholds, use the **addr-trap name set** method (or include the threshold settings while creating the trap). For example:

```
nrcmd> addr-trap trap-1 create
nrcmd> addr-trap trap-1 set low-threshold
nrcmd> addr-trap trap-1 set high-threshold
```

- Step 4** Edit the created scope to which you want to apply the threshold settings. In the SNMP Trap Settings attributes, enter the name of the trap in the *free-address-config* attribute field (see [Figure 19-5](#)). Click **Modify Scope**.

Figure 19-5 SNMP Trap Settings Attributes on the Edit DHCP Scope Page (Local)

Attribute	Value	Data Type	Default	Unset?
free-address-config	trap-1	name reference		<input type="checkbox"/>

Apply Template [none] v

Modify Scope Unset Fields Cancel

149410

In the CLI, use **scope name set free-address-config=trap-name**. For example:

```
nrcmd> scope scope-1 set free-address-config=trap-1
```

When setting the threshold values, it is advisable to maintain a small offset between the low and high values, as described in the “[Simple Network Management](#)” section on page 1-2). The offset can be as little as 5%, for example, a low value of 20% and a high value of 25%, which are the default values.

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.
- Set the traps at the network level or selection tags level.

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.

Network Registrar moves the leases on the removed scope to an orphaned leases pool. When creating a scope, orphaned leases are associated with appropriate scopes.

Removing Scopes if Not Reusing Addresses

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 that you are not immediately planning to re-use the addresses in the scope, then use `scope name delete` to delete it.

Removing Scopes if Reusing Addresses

If you want to re-use the addresses for a scope you want to remove, you have two alternatives:

- 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, deactivate it. See the “[Deactivating Scopes](#)” section on page 19-17. Unlike a removed scope, the server refuses all clients’ renewal requests, which forces many of them to request a new lease. This moves these clients more quickly off the deactivated lease than for a removed scope.

You can use the `ipconfig` utility in Windows to cause a client to release (`/release`) and re-acquire (`/renew`) its leases, thereby moving it off a deactivated 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 deactivated leases in a scope.

Managing DHCP Networks

When you create a scope, you also create a network based on its subnet and mask. Scopes can share the same subnet, so that it is often convenient to show their associated networks and the scopes. 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 Web UI, click **DHCP**, then **Networks** to open the View Network Tree page. On this page, you can:

- List the networks—The networks appear alphabetically by name and identify their subnet and any assigned scope-selection tags. Click the plus (+) 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.

-
- Step 1** In the local Web UI, click **DHCP**, then **Networks** to open the List Networks page.
 - Step 2** Click the name of the network you want to edit. This opens the Edit Network page.
 - Step 3** Edit the network data.
 - Step 4** Click **Modify Network**.
-