CHAPTER 21

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 20, "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.

Related Topics

Configuring DHCP Servers

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

- The DHCP server IP address.
- One or more scopes (see the "Defining and Configuring Scopes" section on page 21-2).

Related Topics

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 23-3.

Configuring DHCP Server Interfaces

To configure the DHCP server, accept the Cisco Prime Network Registrar 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 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.

Local Advanced Web UI

Step 1 Choose **Manage Servers** from **Administration > Tasks** to open the Manage Servers page.

Step 2 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.

Step 3 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.

Step 4 Clicking the name of the configured interface opens the Edit DHCP Server Network Interface page, where you can change the address and ports (in Expert mode) of the interface.

Step 5 Click **Modify Interface** when you are done editing.

Step 6 Click **Return** to return to the Manage Servers page.

CLI Commands

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 network interfaces, so use this command to be more specific about which ones to use.

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. (For more on listing leases and defining lease reservations for a scope, see Chapter 23, "Managing Leases.")

Related Topics

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 (see the "Using Expressions in Scope Templates" section on page 21-4).

Local Advanced and Regional Web UI

Scope templates you add or pull from the local clusters are visible on the List DHCP Scope Templates page. To get there, choose **Scope Templates** from **DHCP > DHCP v4**. This functionality is available only to administrators assigned the dhcp-management subrole of the regional central-cfg-admin or local ccm-admin role.

To explicitly create a scope template, click **Add DHCP Scope Template** on this page. This opens the Add DHCP Scope Template page, which includes a number of fields and settings. 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 values (see the "Create a Scope Template" section on page 5-41 that describes these fields).

Related Topics

CLI Commands

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
```

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 30, "Using Extension Points") are used to modify request or response packets.

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.

Table 21-1 lists the scope expression functions. Note that these functions are not case-sensitive.

Table 21-1 Expression Functions

Expression Function	Description
Context Variables	
bcast-addr	Derived from the broadcast address in the subnet, such as 192.168.50.255. Use in any expression field.
first-addr	Derived from the first address in the subnet, such as the first address in 192.168.50.64/26 is 192.168.50.65. Use in any expression field.
last-addr	Derived from the last address in the subnet, such as the last address in 192.168.50.64/26 is 192.168.50.127. Use in any expression field.
mask-addr	Derived from the network mask address in the subnet, such as 255.255.255.0. Use in any expression field.
mask-count	Derived from the number of bits in the network address of the subnet, such as 24. Use in the Scope Name Expression or Embedded Policy Option Expression field.
naddrs	Derived from the number of IP addresses in the subnet, such as 255. Use in the Scope Name Expression field.
nhosts	Derived number of usable hosts in the subnet, such as 254. Use in any expression field.
subnet	Derived from the IP address and mask of the subnet, such as 192.168.50.0/24. Use in the Scope Name Expression or Embedded Policy Option Expression field.
subnet-addr	Derived from the subnet address, such as 192.168.50.0. Use in any expression field.
template.*attribute*	Attribute of the scope template, such as template.ping-timeout. Use in the Embedded Policy Option Expression field.
Arithmetic Operations (unsigned integer arguments only)	
(+ *arg1 arg2***)**	Adds the two argument values, such as (+ 2 3).
(– *arg1 arg2***)**	Subtracts the second argument value from the first one, such as with *ping-timeout* defined as 100, (– template.ping-timeout 10) yields 90.
(* *arg1 arg2***)**	Multiplies the values of two arguments.

Table 21-1 Expression Functions (continued)

Expression Function	Description
(/ *arg1 arg2***)**	Divides the value of the first argument by that of the second one (which cannot be zero).
Concatenation Operation	

(concat *arg1 ... argn***)**	Concatenates the arguments into a string, to be used in the Scope Name Expression field. Examples: With subnet=192.168.50.0/24 and template.ping-timeout=100:
	`(concat "ISP-" subnet) --> ISP-192.168.50.0/24`
	`(concat subnet "-" (+ template.ping-timeout 10))`
	`--> 192.168.50.0/24-110`
	`(concat "ISP-" subnet "-" (+ template.ping-timeout 10))`
	`--> ISP-192.168.50.0/24-110`
	See also the "Scope Name Expression Example" section on page 21-7.

Create Option Operation	

(create-option *opt val***)**	Use create-option in the Embedded Policy Option Expression field to create new DHCP options for the scope. The first argument can be an integer or string to represent the option number or name. The second argument can be a string or blob to give the option a value.
	You can also specify custom defined and unknown options. For undefined options, the option number must be specified and the data is used as is (as blob data). If the data is a string, the string is used as is and if the data is a number or address, it is used as is.
	Examples:
	`(list (create-option "domain-name" "example.com")`
	`(create-option 3 "10.10.10.1"))`
	`(create-option "routers" "10.10.10.1,10.10.10.2,10.10.10.3")`
	`(create-option "routers" (create-ipaddr subnet 10))`
	See also the "Embedded Policy Option Expression Example" section on page 21-7.

Create Vendor Option Operation	

(create-vendor-option *set-name opt val***)**	Use the create-vendor-option in the Embedded Policy Option Expression field to creates a DHCP vendor option. The *set-name* specifies the option definition set for the vendor option. The *opt* can be the literal string or integer identifying the vendor option in the set. The *val* is representation of the option value.
	For example:
	`(list (create-option "routers" (create-ipaddr subnet 1))`
	`(create-vendor-option "dhcp-cablelabs-config" 125`
	`(concat "(tftp-servers 2 " (create-ipaddr subnet 2) ")")))`

Table 21-1 Expression Functions (continued)

Expression Function	Description
Create Range Operation	

(create-range *start end***)**	Use this operation in the Range Expression field. It creates an IP address range for the scope. The first argument is the start of the address range and can be an integer or IP address string. The second argument is the end of the range and can be an integer or IP address string. Do not include the local host or broadcast address determined by the mask (such as 0 and 255 for /24 subnets) in the range. Validation ensures that the range must be in the subnet defined by the template and that the first argument value must be lower than the second. An integer value determines the position of the address in the given subnet. Examples (with subnet=192.168.50.0/26):
	`(create-range "192.168.50.65" "192.168.50.74")`
	`--> 192.168.50.65 - 192.168.50.74`
	`(create-range 1 10) --> 192.168.50.65 - 192.168.50.74`
	See also the "Range Expression Example" section on page 21-7.

Create IP Operation	
(create-ipaddr *net host***)**	Use this operation in the Embedded Policy Option Expression or Range Expression fields. It creates an IP address string. The net argument is a string or variable. The host argument is an integer. Example: `(create-ipaddr subnet 4)`
List Operation	

(list *oper1 ... opern***)**	Arguments must all be create-option or create-range operations. Nesting is not supported. Examples:
	`(list (create-option "routers" "10.10.10.1")`
	`(create-option "domain-name" "example.com"))`
	`(list (create-range 1 5) (create-range 10 20))`

Local Advanced and Regional Web UI

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

- **Scope Name Expression**—Must return a string
- **Range Expression**—Must return IP addresses
- **Embedded Policy Option Expression**—No requirements

CLI Commands

Use the following **scope-template** command attributes:

- **scope-name**
- **ranges-exp**
- **options-exp**

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 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.

To set the range expressions in the CLI, you should place the expression into a file and use a command such as:

```
nrcmd> scope-template example-template set ranges-expr=@file

where file is the name of the file that you created with the expressions.
```

Embedded Policy Option Expression Example

An embedded policy is important because the DHCP server looks at it before it looks at the assigned, named policy of the scope. 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 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.

To set the policy options expressions in the CLI, you should place the expression into a file and use a command such as:

```
nrcmd> scope-template example-template set options-expr=@file
```

where *file* is the name of the file that you created with the expressions.

Note Trying to specify the expression directly on the CLI command line will likely fail because of embedded spaces and special characters such as the quotes. Use the *@file* syntax as it avoids any potential issues with the CLI command parser. But the Web UI does not support the @file syntax. You can enter complex expressions directly in the Web UI.

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 (see the "Creating and Applying Scope Templates" section on page 21-3) except for an additional attribute unset function. Make your changes, then click **Modify Scope Template**.

In the CLI, edit a scope template attribute by 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.

Caution Be careful applying a scope template to an existing scope. The template overwrites all the scope attributes with its own, which can have a detrimental effect if the scope is active.

Local Advanced Web UI

- **While creating a named scope**—On the List/Add DHCP Scopes page, include the name of the scope, add its subnet and mask, then choose the scope template from the drop-down list. Clicking **Add Scope** creates a scope with the name specified and with the attributes set for the scope template, including the expressions you might have set (see the "Using Expressions in Scope Templates" section on page 21-4). (Note that Basic mode lets you specify a Class of Service, but not apply a scope template.)

- **When a template is applied to a target**—if the scope-template has an embedded policy, it is copied to the scope. This embedded policy may or may not have options. As the entire scope-template's embedded policy is used (if it exists), it will wipe out any existing options in the scope. If the scope-template has no embedded policy, the scope's embedded policy is retained. Next the scope-template's option expression, if any, is evaluated and the options are added to the embedded policy options in the scope (if no embedded policy exists, one is created).
- **While creating a scope, derive its name from the template**—If you set a Scope Name Expression for the scope template (see the "Using Expressions in Scope Templates" section on page 21-4) on the Add DHCP Scope Template page, when you add a scope on the List/Add DHCP Scopes page, omit the name of the scope, but add its subnet and mask, then choose the scope template from the Template drop-down list. Clicking **Add Scope** 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. (Note that Basic mode does not provide this functionality.)
- **After creating a named scope**—On the Edit DHCP Scopes page, scroll to the bottom to find the **Apply Template** button. Choose a preconfigured template from the drop-down list, then click the button. Then click **Modify Scope**. (Be aware of the previous warning that the template attributes overwrite the existing ones of the scope.)

CLI Commands

To apply a template to the scope while creating the scope, use **scope** *name* **create** *address mask* **template=***template-name*. For example:

```
nrcmd> scope example-scope create 192.168.50.0 24 template=example-scope-template
```

To derive the scope name from the template during scope creation, use **scope-template** *name* **apply-to** {**all** | *scope1*,*scope2*,...}. For example:

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

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:

```
nrcmd> 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

You can configure scopes at the local cluster only. The web UI pages are different for local basic and advanced modes.

Local Basic Web UI

Step 1 Choose **Scopes** from **DHCP > DHCP v4** to open the Manage Scopes page.

Step 2 Choose a VPN for the scope, if necessary.

Step 3 Enter a scope name, enter the subnet IP address and choose a mask value from the drop-down list.

Step 4 If desired, choose a preconfigured class of service (client-class) for the scope from the drop-down list.

Step 5 Click **Add Scope**.

Step 6 Reload the DHCP server.

Local Advanced Web UI

Step 1 Choose **Scopes** from **DHCP > DHCP v4** to open the List/Add DHCP Scopes page.

Step 2 Choose a VPN for the scope, if necessary.

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 Expressions in Scope Templates" section on page 21-4). In the latter case, choose the scope template. You must always enter a subnet and mask for the scope.

Step 4 Click **Add Scope**. This opens the 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 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). Add the range, then click **Add Range**.

Step 7 Click **Add Scope**.

Step 8 Reload the DHCP server.

Tip To view any leases and reservations associated with the scope, see Chapter 23, "Managing Leases." To search for leases, see the "Searching Server-Wide for Leases" section on page 23-9.

Related Topics

Getting Scope Counts on the Server, page 21-10
Configuring Multiple Scopes, page 21-11
Editing Scopes, page 21-17
Staged and Synchronous Mode, page 21-18

Getting Scope Counts on the Server

You can view the created scopes associated with the DHCP server, hence obtain a count, in the web UI.

CLI Commands

Using the CLI, you can get an exact count of the total scopes for the DHCP server by using **dhcp getScopeCount** [**FailoverPair** *name* | **vpn** *name* | **all**]. You can specify a VPN or all VPNs. Omitting the **vpn** *name* returns a count for the current VPN. Specifying a failover pair name returns the total scopes and networks for the failover pair. Because a failover pair definition includes explicit VPN settings in its matchlist, these counts are not limited to the current VPN only.

To create a scope, use **scope** *name* **create**. Each scope must identify its network address and mask. When you create the scope, Cisco Prime Network Registrar places it in its current virtual private network (VPN), as defined by **session set current-vpn**. You cannot change the VPN once you set it at the time of creation of the scope.

To set a policy for the scope, use **scope** *name* **set policy**.

To add a range of IP addresses to the scope, use **scope** *name* **addRange**.

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 21-12).

Configuring the addresses of a a single subnet 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.

Related Topics

Configuring Multiple Scopes for Round-Robin Address Allocation, page 21-11
Configuring Multiple Scopes Using Allocation Priority, page 21-12

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 Cisco Prime 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 28, "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 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 accessed 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.

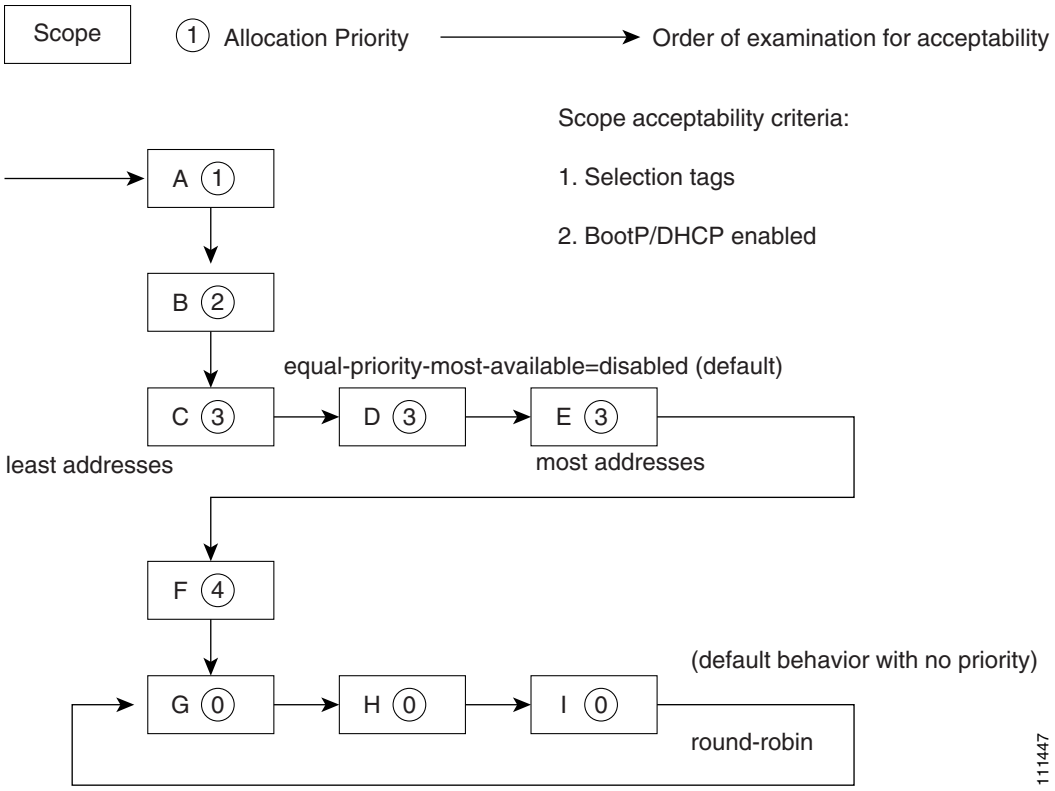
When there is more than one scope in a network, then the DHCP must decide which scope to allocate an IP address from when it processes a DHCPDISCOVER request from a DHCP client that is not already associated with an existing address. The algorithm that the DHCP server uses to perform this allocation 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 21-1 shows an example of a network with nine scopes (which is unusual, but serves to illustrate several possibilities of using allocation priority).

Figure 21-1 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) 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 21-1. 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 25-13), 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 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 21-2.

Table 21-2 Address Allocation Priority Settings

Attribute	Type	Description
allocation-priority	Scope (set or unset)	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 preset value) 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. If set, this attribute overrides the DHCP server *priority-address-allocation* attribute setting. However, if *allocation-priority* is unset and *priority-address-allocation* is enabled, then the allocation priority for the scope is its subnet address. With *allocation-priority* unset and *priority-address-allocation* disabled, the scope is examined in the default round-robin fashion.
allocate-first-available	Scope (enable or disable)	If enabled, forces all allocations for new addresses from this scope to be from the first available address. If disabled (the preset value), uses the least recently accessed address. If set, this attribute overrides the DHCP server *priority-address-allocation* attribute setting. However, if unset and *priority-address-allocation* is enabled, then the server still allocates the first available address. With *allocate-first-available* unset and *priority-address-allocation* disabled, the scope is examined in the default round-robin fashion.

Table 21-2 Address Allocation Priority Settings (continued)

Attribute	Type	Description
failover-backup-allocation-boundary	Scope (set or unset)	If *allocate-first-available* 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. 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. See Figure 21-2 on page 21-16 for an illustration of how addresses are allocated in a scope using this setting.
priority-address-allocation	DHCP (enable or disable)	Provides a way to enable priority address allocation for the entire DHCP server without having to configure it on every scope. (However, the scope *allocation-priority* setting overrides this one.) If *priority-address-allocation* is enabled and the scope *allocation-priority* attribute is unset, then the scope subnet address is used for the allocation priority. If the scope *allocate-first-available* 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.
equal-priority-most-available	DHCP (enable or disable)	By default, when two or more scopes with the same nonzero *allocation-priority* 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 *equal-priority-most-available* 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.

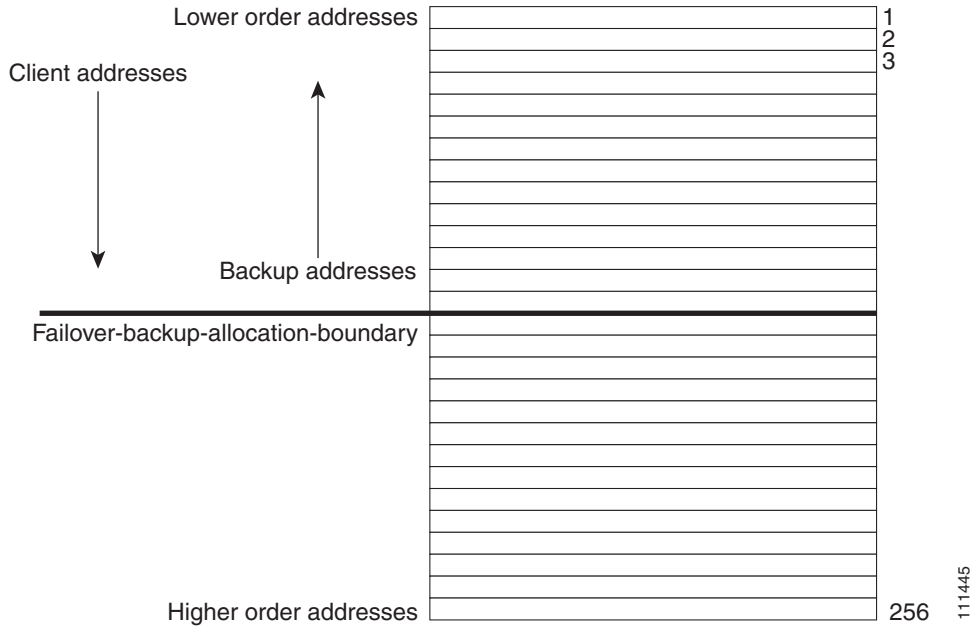
Allocating Addresses In Scopes

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 accessed address first, from the list of available leases. But all the operations that require accessing the lease like listing all the leases or all leases in a scope, asking for a specific lease (nrcmd> *lease addr*), searching leases, or modifying leases (activate, deactivate, or force available) affect the ordering of the leases in the list of available leases with the server.

Operating on a single lease places that lease at the end of the list. Listing leases causes the leases to be arranged in numerical order, making the lowest numbered lease to end up first on the available list. Other operations that require the server to access the lease, like leasequery requests also impacts the order of leases.

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 21-2.

Figure 21-2 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 want 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 addresses. However, if you configure *allocate-first-available*, then the allocated addresses tend to cluster low in the scope 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

Note You can only make changes to a scope's subnet, if there are no reservations or ranges that conflicts with the change, either in the current scope or any other scope with the same old subnet as those scopes' subnet will also be changed.

Local Advanced Web UI

Step 1 Create a scope, as described in the "Creating Scopes" section on page 21-9.

Step 2 Reload the DHCP server.

Step 3 Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page. (If a server reload is required, a status message indicates it and you must reload first before proceeding.)

Step 4 Modify the fields or attributes as necessary. You can also modify the name of the scope.

Step 5 To edit the scope embedded policy, see the "Configuring Embedded Policies for Scopes" section on page 21-18. To list leases for the scope, see the "Viewing Leases" section on page 23-2.

Step 6 Click **Modify Scope**.

Step 7 Reload the DHCP server.

CLI Commands

After you create a scope, 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*). Then:

- To reset an attribute, use **scope** *name* **set**. For example, you can reset the name of the scope by using **scope** *name* **set** *name=new name*
- To enable or disable an attribute, use **scope** *name* **enable** or **scope** *name* **disable**.

See the **scope** command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions.

Related Topics

Staged and Synchronous Mode, page 21-18
Configuring Embedded Policies for Scopes, page 21-18
Configuring Multiple Subnets on a Network, page 21-19
Enabling and Disabling BOOTP for Scopes, page 21-20
Disabling DHCP for Scopes, page 21-20
Deactivating Scopes, page 21-21
Setting Scopes to Renew-Only, page 21-21
Setting Free Address SNMP Traps on Scopes, page 21-21
Removing Scopes, page 21-22

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 (including deletions) 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 you add or modify a scope while in staged mode and then change the dhcp edit mode to synchronous, the first change in synchronous mode applies all pending changes for that scope (not just the ones made while in synchronous mode).

Note In Cisco Prime Network Registrar versions earlier than Release 7.1, the dhcp edit mode was called scope edit mode.

Local Basic or Advanced Web UI

To view the current dhcp edit mode or change the dhcp edit mode, click the *username* drop-down list on the top right of the window and choose **Session Settings**. If the scope is up to date in the DHCP server, the `Total synchronized scopes` message appears on the List/Add DHCP Scopes page (in Advanced mode) and the `Scope status: synchronized` message appears on the Edit DHCP Scope page (in both modes). If the scope is not up to date, the `Scope` *`name`* `status: reload required` message is displayed.

CLI Commands

View the dhcp edit mode by using **session get dhcp-edit-mode**, or set the dhcp edit mode using **session set dhcp-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, Cisco Prime 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 22-1 describes, the DHCP server looks at the embedded policy of a scope before it looks at its assigned, named policy.

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

Local Advanced Web UI

Step 1 Create a scope, as described in the "Creating Scopes" section on page 21-9.

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 **Create New Embedded Policy** to create a new embedded policy, or **Edit Existing Embedded Policy** if there is already an existing one, 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**.

CLI Commands

Create a scope first. In the CLI, **scope-policy** uses the same syntax as **policy**, except that it takes the scope name as an argument. Then, to:

- Determine if there are any embedded property values already set for a scope, use **scope-policy** *scope-name* **show**.
- Enable or disable an attribute, use **scope-policy** *name* **enable** or **scope-policy** *name* **disable**.
- Set and unset attributes, use **scope-policy** *name* **set** and **unset**.
- List, set, and unset vendor options (see the "Using Standard Option Definition Sets" section on page 22-8).

Configuring Multiple Subnets on a Network

Cisco Prime 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.

Local Advanced Web UI

Step 1 Create a scope (see the "Creating Scopes" section on page 21-9) that you will 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 To make this a secondary scope, enter the network address of the subnet of the primary scope in the *Primary Subnet* attribute field in the Edit DHCP Scope page.

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 Cisco Prime 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.

Step 4 Click **Modify Scope**.

Step 5 Restart or reload the server.

CLI Commands

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.

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 Cisco Prime 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 reuse its lease simply by forcing lease availability. See the "Forcing Lease Availability" section on page 23-19.

Local Advanced Web UI

On the Edit DHCP Scope page, under BootP Settings, enable the *bootp* attribute for BOOTP, or the *dynamic-bootp* attribute for dynamic BOOTP. They are disabled by default. Then click **Modify Scope**.

CLI Commands

Use **scope** *name* **enable bootp** to enable BOOTP, and **scope** *name* **enable dynamic-bootp** to enable dynamic BOOTP. Reload the DHCP server (if in staged dhcp edit mode).

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 21-20. 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 21-21.

Local Advanced Web UI

On the Edit DHCP Scope page, under BootP Settings, disable the *dhcp* attribute and enable the *bootp* attribute. Then click **Modify Scope**.

CLI Commands

Use **scope** *name* **disable dhcp** to disable DHCP. You should also enable BOOTP and reload the server (if in staged dhcp edit mode).

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.

Local Advanced Web UI

On the Edit DHCP Scope page, under Miscellaneous Settings, explicitly enable the *deactivated* attribute. Then click **Modify Scope**.

CLI Commands

Use **scope** *name* **enable deactivated** to disable BOOTP and DHCP for the scope. Reload the DHCP server (if in staged dhcp edit mode).

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.

Local Advanced Web UI

On the Edit DHCP Scope page, under Miscellaneous Settings, explicitly enable the *renew-only* attribute. Then click **Modify Scope**.

CLI Commands

Use **scope** *name* **enable renew-only** 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. You can also set traps based on networks and selection tags instead of scopes.

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 1-4). The offset can be as little as 5%, for example, a low value of 20% and a high value of 25%, which are the preset 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.

Local Advanced Web UI

Step 1 Create a trap configuration by choosing **Traps** from the **DHCP** drop-down list to open the List/Add Trap Configurations page.

Step 2 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 AddrTrapConfig**. (You can go back to edit these values if you need to.)

Step 3 Edit the created scope to which you want to apply the threshold settings. Under SNMP Trap Settings, enter the name of the trap in the *free-address-config* attribute field. Click **Modify Scope**.

CLI Commands

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
```

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

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

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 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 reused 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.

Cisco Prime 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.

Related Topics

Removing Scopes if Not Reusing Addresses, page 21-23
Removing Scopes if Reusing Addresses, page 21-23

Removing Scopes if Not Reusing Addresses

You can remove scopes if you are not reusing addresses.

Local Basic or Advanced Web UI

If you are sure you do not plan to reuse the scope, on the Manage Scopes or List/Add DHCP Scopes page, click the Delete icon next to its name, and confirm or cancel the deletion.

CLI Commands

Be sure that you are not immediately planning to reuse the addresses in the scope, then use **scope** *name* **delete** to delete it.

Removing Scopes if Reusing Addresses

If you want to reuse 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 reuse 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 21-21. 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.

Related Topics

Listing Networks, page 21-24
Editing Networks, page 21-24

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 Basic mode, choose **DHCP v4 > Networks** from the **DHCP** to open the DHCP Network Tree page (for DHCP v6 networks, this opens the DHCP v6 Networks page). On this page, you can:

- **List the networks**—The networks appear alphabetically by name. You can identify their subnet and any assigned selection tags. Click the plus (+) sign next to a network to view the associated scopes.

  To expand all network views, click **Expand All**. To collapse all network views to show just the network names, click **Collapse All**.
- **Edit a network name**—Click the network name. See "Editing Networks" section on page 21-24.

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.

Local Basic or Advanced Web UI

Step 1 Choose **Networks** from **DHCP > DHCP v4 or DHCP > DHCP v6** to open the DHCP Network Tree page (DHCP v4) or the DHCP v6 Networks (DHCP v6).

For DHCPv6, the DHCP v6 Networks page is for creating networks. Enter a name for the network, choose a template, if desired, and enter the template root prefix name and click **Add Link** (see the "Viewing DHCPv6 Networks" section on page 27-32.).

If you want to edit a network, click the name of the network you want to edit. This opens the Edit DHCP v6 Link page.

Step 2 Click **Modify Network**.