# Managing Policies and Options

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 DHCPv4 scope (dynamic address pool) or DHCPv6 prefix to the server. The policy attributes and options are assigned to the scope or prefix.

# Configuring DHCP Policies

Every DHCPv4 scope or DHCPv6 prefix must have one or more policies defined for it. Policies define lease duration, gateway routers, and other configuration parameters, in what are called DHCP options. Policies are especially useful if you have multiple scopes or prefixes, because you need only define a policy once.

This section describes how you can define named policies with specific attributes and option definitions, or use system default or embedded policies.

## Related Topics

Types of Policies, on page 3

DHCPv4 Policy Hierarchy, on page 4

Creating and Applying DHCP Policies, on page 6

Cloning a Policy, on page 8

Setting DHCP Options and Attributes for Policies, on page 9

Creating and Editing Embedded Policies, on page 11

# Configuring DHCPv6 Policies

You can edit DHCPv6 policy attributes, which are:

- ***affinity-period***—See Lease Affinity (no preset value).
- ***allow-non-temporary-addresses***—Enable or disable DHCPv6 clients requesting nontemporary (IA_NA) addresses (preset value enable).
- ***allow-rapid-commit***—With Rapid Commit enabled, clients receive information (when solicited) on committed addresses, which are then more quickly committed with a client request (preset value disable). Use Rapid Commit only if one DHCP server is servicing clients, otherwise it might seem like the client is receiving multiple addresses. (See DHCPv6 Policy Hierarchy, on page 5 for special handling of this attribute, and Reconfigure support, when used in an embedded or named policy for a prefix.)
- ***allow-temporary-addresses***—Enable or disable DHCPv6 clients requesting temporary (IA_IA) addresses (preset value enable).
- ***default-prefix-length***—For prefix delegation, default prefix length of the delegated prefix if the client or router does not explicitly request it; must always be less than or equal to the prefix range prefix length (preset value 64 bytes).
- ***reconfigure***—Enables special handling during the policy hierarchy processing when checking the Prefix policies (embedded or named) for the Prefixes on a Link (see Reconfiguring IPv6 Leases).
- ***preferred-lifetime***—Default and maximum preferred lifetime for leases (preset value 1 week).
- ***v6-reply-options***—DHCPv6 options returned in replies to clients (no preset value). (See DHCPv6 Policy Hierarchy, on page 5 for special handling of this attribute when used in an embedded or named policy for a prefix.)
- ***valid-lifetime***—Default and maximum valid lifetime for leases (preset value 2 weeks).

**Tip** For details on the Reconfigure attributes, see Reconfiguring IPv6 Leases.

# Reconfigure Support (DHCPv6)

For DHCPv6, a server can send a RECONFIGURE message to a client to inform the client that the server has new or updated configuration parameters. If so authorized and through proper authentication, the client then immediately initiates a Renew, Rebind, or Information-request reply transaction with the server so that the client can retrieve the new data. Without this support, a client must wait until it renews its lease to get configuration updates.

You can have the server unicast the Reconfigure packet or deliver it through a relay agent. If you do not specify either way, the client's client-class policy, requested lease's prefix or link policies, or system_default_policy (but not the client policy) determines the preferred method. If the unicast method is not available (the client has no valid address lease), the server uses the relay agent; with no relay agent, the server tries to unicast; failing both results in an error. With the unicast method, if the specified lease is not usable, the server selects the lease with the longest valid lifetime.

The server and client negotiate Reconfigure support through the added security of a reconfigure key. The internal process is basically:

1. The client sends the server a REQUEST, SOLICIT, or ADVERTISE packet that includes the *reconfigure-accept* option (20) to indicate that the client wants to accept Reconfigure messages.

(Conversely, the DHCP server can send a *reconfigure-accept* option to the client about whether the client should accept Reconfigure messages.) This option is required for Reconfigure support.

2. If the Cisco Prime Network Registrar policy for the client has the *reconfigure* attribute set to **allow** or **require** (rather than **disallow**), the DHCP server accepts the packet and generates a reconfigure key for the client. (The server records the key value and its generation time in the *client-reconfigure-key* and *client-reconfigure-key-generation-time* attributes for the DHCPv6 lease.)
3. The server sends a Reply packet to the client with the reconfigure key in the *auth* option (11) along with the *reconfigure-accept* option.
4. The client records the reconfigure key to authenticate Reconfigure messages from the server.
5. When the server wants to reconfigure the client, it sends a Reconfigure packet with the *reconfigure-message* option (19) and an *auth* option containing a hash generated from the packet and the reconfigure key. The *reconfigure-message* option indicates in the *msg-type* field whether the client should respond with a Renew or an Information-request packet.
6. Upon receiving the packet, the client validates that the *auth* option contains the valid hash, then returns a Renew, Rebind, or Information-request packet. This packet includes an Option Request (*oro*) option (6) to indicate specific option updates. (If the server does not receive a reply from the client in a preconfigured timeout value of 2 seconds, the server retransmits the Reconfigure message at most 8 times, then aborts the reconfigure process for the client.)
7. The server sends the client a Reply packet that includes options for configuration parameters. The packet might also include options containing addresses and new values for other configuration parameters, even if the client did not request them. The client records these changes.

# Types of Policies

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

- **System default (system_default_policy)**—Provides a single location for setting default values on certain options for all scopes or prefixes. Use the system default policy to define attributes and standard DHCP options that have common values for all clients on all the networks that the DHCP server supports. You can modify the system default options and their values. If you delete a system default policy, it reappears using its original list of DHCP options and their system-defined values (see the table below)

*Table 1: System Default Policy Option Values*

| System Default Option | Predefined Value |
|---|---|
| all-subnets-local | False |
| arp-cache-timeout | 60 seconds |
| broadcast-address | 255.255.255.255 |
| default-ip-ttl | 64 |
| default-tcp-ttl | 64 |
| dhcp-lease-time | 604800 seconds (7d) |
| ieee802.3-encapsulation | False |
| interface-mtu | 576 bytes |

| System Default Option | Predefined Value |
|---|---|
| mask-supplier | False |
| max-dgram-reassembly | 576 bytes |
| non-local-source-routing | False |
| path-mtu-aging-timeout | 6000 seconds |
| path-mtu-plateau-tables | 68, 296, 508, 1006, 1492, 2002, 4352, 8166, 17914, 32000 |
| perform-mask-discovery | False |
| router-discovery | True |
| router-solicitation-address | 224.0.0.2 |
| tcp-keepalive-garbage | False |
| tcp-keepalive-interval | 0 seconds |
| trailer-encapsulation | False |

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

  Cisco Prime Network Registrar includes a policy named **default** when you install the DHCP server. The server assigns this policy to newly created scopes and prefixes. You cannot delete this default policy.

- **Embedded**—A policy embedded in (and limited to) a named scope, scope template, prefix, prefix template, client, or client-class. An embedded policy is implicitly created (or removed) when you add (or remove) the corresponding object. Embedded policy options have no default values and are initially undefined.

**Tip** Be sure to save the object (scope, prefix, client, or client-class) for which you are creating or modifying an embedded policy. Not doing so is a common error when using the web UI. Click **Modify** for both the embedded policy and the parent object.

# Policy Hierarchy

## DHCPv4 Policy Hierarchy

To eliminate any conflicting attribute and option values that are set at various levels, the Cisco Prime Network Registrar DHCP server uses a local priority method. It adopts the more locally defined attribute and option values first while ignoring the ones defined on a more global level, and includes any default ones not otherwise defined. When the DHCP server makes processing decisions for a DHCPv4 client, it prioritizes the attributes and options in this order:

1. Client embedded policy.
2. Client named policy.
3. Client-class embedded policy.
4. Client-class named policy.
5. Scope embedded policy for clients, or address block embedded policy for subnets.
6. Scope named policy for clients (or default policy if a named policy is not applied to the scope), or address block named policy for subnets.
7. Any remaining unfulfilled attributes and options in the system_default_policy. For attributes, the default value for the most local policy applies.

**Note** For DHCPv6 policy prioritization, see DHCPv6 Policy Hierarchy, on page 5.

# DHCPv6 Policy Hierarchy

DHCPv6 uses the existing policy objects, with additional DHCPv6 specific attributes (that are mostly analogous to those in DHCPv4). For DHCPv6, the hierarchy is:

1. Client embedded policy
2. Client named policy
3. Client-class embedded policy
4. Client-class named policy
5. Prefix embedded policy
6. Prefix named policy
7. Link embedded policy
8. Link named policy
9. system_default_policy

For attributes, the default value for the most local policy applies. This hierarchy is the same as for DHCPv4, except for the additional link policies and the fact that the prefix policies replace the scope policies. (For a comparison with the DHCPv4 policy hierarchy, see DHCPv4 Policy Hierarchy, on page 4.)

The hierarchy applies to most policy attributes, which the server processes in the context of a single prefix. However, the server processes a few attributes (specifically *allow-rapid-commit*, *reconfigure*, *v6-reply-option*, *v6-options*, and *v6-vendor-options*) in the context of multiple prefixes. In these cases, the processing at the prefix levels (steps 5 and 6) is a bit different:

- For the *reconfigure* attribute that controls whether the server requires, allows, or disallows client reconfiguration, the server checks the embedded and named policies of all prefixes on the link that the client is allowed to use (based on selection tags). If any of the prefix policies have the *reconfigure* attribute set to **disallow** or **require**, the server uses that setting. Otherwise, if at least one policy has it set to **allow**, Reconfigure is allowed. Otherwise, the server checks the remaining policies in the hierarchy. (See the Reconfiguring IPv6 Leases for details.)
- If the client requests Rapid Commit (see the Editing DHCPv6 Server Attributes), the server checks the embedded and named policies of all prefixes on the link that the client is allowed to use (based on selection tags). If one of these policies has *allow-rapid-commit* disabled, the server processes the client request as if Rapid Commit were not part of the request. If at least one policy has *allow-rapid-commit* enabled, the

client can use Rapid Commit. If no prefix policy has the attribute configured, processing continues at step 7.

- For the options-related attributes (see Setting DHCPv6 Options, on page 20), the server also does special handling at steps 5 and 6. The server checks the embedded and then named policy of each prefix on the link. It then uses the first one with the configured *v6-reply-option* attribute, or the first one with the configured value for the *v6-options* or *v6-vendor-options*.
- The server checks the prefixes in case-insensitive alphabetical order.
- The server ignores any policies related to the location-independent and/or universal link and the prefixes under those. Only topological links (and prefixes under those links) are considered.

**Tip** In configurations with multiple prefixes on a link, avoid setting the Rapid Commit and option properties for the prefix policy, but rather set them on the link policy or other policy instead.

# Creating and Applying DHCP Policies

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

- **Name**—Not case-sensitive and must be unique.
- *permanent-leases* **attribute**—A permanent lease never expires.
- **Lease time**—How long a client can use an assigned lease before having to renew the lease with the DHCP server (the lease time attributes are not available for an embedded policy, only the option). The default lease time for both system default and default policies is seven days (604800 seconds). A policy contains two lease times—the client lease time and the server lease time:
    - **Client lease time**—Determines how long the client believes its lease is valid. (Set the client lease time using a DHCP option, not a policy attribute.)
    - **Server lease time**—Determines how long the server considers the lease valid. Note that the server lease time is independent of the lease grace period. The server does not allocate the lease to another client until after the lease time and grace period expire.

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

  You can establish these two different lease times if you want to retain information about client DNS names and yet have them renew their leases frequently. When you use a single lease time and it expires, the server no longer keeps that client DNS name. However, if you use a short client lease time and a longer server lease time, the server retains the client information even after the client lease expires. For details on leases, see Managing Leases.

- **Lease grace period**—Time period after the lease expires that it is unavailable for reassignment (not available for an embedded policy).
- **DNS update configuration**—A DNS update configuration specifies the type of DNS updates to perform, the zones involved, the DNS server to be updated, and the related security. The policy determines the forward and reverse DNS update configuration objects, and can also specify the forward zone to use if

a DNS server hosts multiple zones. (For details on DNS update configurations, see Creating DNS Update Configurations.)

- **DHCP options**—To add option values, see Setting DHCP Options and Attributes for Policies, on page 9.

## Local Basic or Advanced and Regional Web UI

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

**Step 2** The default policy and system_default_policy are already provided for you.

**Step 3** Click the **Add Policies** icon in the Policies pane, give the policy a unique name (required).

**Step 4** Set the offer timeout and grace period values or leave them blank.

**Step 5** Enter the DHCP Lease Time, if required and click **Add DHCP Policy** to add the named policy.

**Step 6** In the Edit DHCP Policy page, you can:

- Add the necessary DHCP options (see Setting DHCP Options and Attributes for Policies, on page 9 like:
  - **Lease time**—Set the dhcp-lease-time (51) option.
  - **Limitation count**—See Using Expressions.
  - **Use client IDs for reservations**—See Overriding Client Identifiers.

  To set vendor-specific options, see Using Standard Option Definition Sets, on page 12.
- In Advanced mode, set the policy attributes, which include:
  - **Unavailable timeout**—See Setting Timeouts for Unavailable Leases.
  - **Inhibit all renews**—See Inhibiting Lease Renewals.
  - **Inhibit all renews at reboot**
  - **Permanent leases** (not recommended)
  - **Lease retention limit**
- Set the DNS update configuration that determines which forward or reverse zones you want to include in a DNS update (**DNS Update Settings**). You can set:
  - ***forward-dnsupdate***—Name of the update configuration for the forward zone. Note that you can thereby set different update configurations for forward and reverse zones.
  - ***forward-zone-name***—If necessary, overrides the forward zone in the update configuration. Use this in case a DNS server is hosting multiple zones.
  - ***reverse-dnsupdate***—Name of the update configuration for the reverse zone. If not set on any policy in the policy hierarchy applicable to the client request (see DHCPv4 Policy Hierarchy, on page 4), the DHCP server uses the forward-dnsupdate configuration.

**Step 7** Click **Save**.

**Step 8** Reload the DHCP server.

In the regional web UI, you can also pull replica policies and push policies to local clusters. (See Configuring DHCP Policies, on page 1 for regional policy management.)

# CLI Commands

Use **policy** *name* **create** to create the policy. Then use **policy** *name* **set offer-timeout=***value* and **policy** *name* **set grace-period=***value* to set these two values.

To set policy options, use **policy** *name* **setOption** <*opt-name* | *id*> *value* [**-blob**] [**-roundrobin**]:

- **Lease time**—Use **policy** *name* **setLeaseTime** *time-val*.
- **Subnet mask**—Use a combination of **policy** *name* **setOption subnet-mask** *value* and **dhcp enable get-subnet-mask-from-policy**.

When **-roundrobin** is enabled, instructs the DHCP server to return option data which contains more than one value in a different, rotated, order. A particular client will always get the same order, but different clients will get different "rotations" of the order of the multiple values configured for the option based on their client identifier.

To confirm the option settings, use **policy** *name* **listOptions** or **policy** *name* **getOption** <*opt-name* | *id*>.

To enable permanent leases (not recommended), use **policy** *name* **enable permanent-leases**. Note that enabling permanent leases forces the *dhcp-lease-time* option (51) to be set to infinite.

When connected to a regional cluster, you can use the following pull, push, and reclaim commands. For push and reclaim, a list of clusters or "all" may be specified.

- **policy** < *name* | **all** > **pull** < **ensure** | **replace** | **exact** > *cluster-name* [**-report-only** | **-report**]
- **policy** < *name* | **all** > **push** < **ensure** | **replace** | **exact** > *cluster-list* [**-report-only** | **-report**]
- **policy** *name* **reclaim** *cluster-list* [**-report-only** | **-report**]

## Related Topics

Types of Policies, on page 3

DHCPv4 Policy Hierarchy, on page 4

Cloning a Policy, on page 8

Setting DHCP Options and Attributes for Policies, on page 9

Creating and Editing Embedded Policies, on page 11

Creating DHCP Option Definition Sets and Option Definitions, on page 11

# Cloning a Policy

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

```
nrcmd> policy cloned-policy create clone=example-policy-1 offer-timeout=4m
```

# Setting DHCP Options and Attributes for Policies

DHCP options automatically supply DHCP clients with configuration parameters, such as domain, nameserver, and subnet router addresses (see Creating DHCP Option Definition Sets and Option Definitions, on page 11). Note that the Cisco Prime Network Registrar user interfaces allow you to set some option values on a policy that actually have no effect on the packet returned to the client (such as *hostname* and *dhcp-server-identifier*).

The server searches the policies, in order, for these BOOTP and DHCP attribute values and returns the first occurrence of these values in its reply packet:

- *packet-siaddr* returned in the *siaddr* packet field
- *packet-file-name* returned in the *file* field
- *packet-server-name* returned in the *sname* field

## Related Topics

Adding Option Values, on page 9

Adding Complex Values for Suboptions, on page 10

# Adding Option Values

You can view, set, unset, and edit DHCP option values. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed for the given option name. Cisco Prime Network Registrar DHCP options are grouped into categories to aid you in identifying options that you must set in various usage contexts. You can create custom option definitions to simplify entering custom option values (see Creating Custom Option Definitions, on page 14).

## Local Basic or Advanced and Regional Web UI

**Step 1** Create a policy, as described in Creating and Applying DHCP Policies, on page 6.

**Step 2** On the Edit DHCP Policy page, add each DHCP option to the policy by choosing its number and name in the drop-down list. The choices indicate the data type of the option value (see Option Definition Data Types and Repeat Counts, on page 20).

**Tip** You can sort the options by Name, Number, or (in the case of DHCPv4) Legacy (grouping).

**Step 3** Add the appropriate option value in the Value field. The web UI does error checking based on the value entered. For example, to add the lease time for the policy, click the *[51] dhcp-lease-time (unsigned time)* option in the Number drop-down list, then add a lease time value in the Value field. (Options do not have preset values.)

**Tip** If you are configuring an option on a policy while another user is editing the option definition, log out of the session and log back in to get the new option definition.

**Step 4** Click **Add Option** for each option. You must supply a value or you cannot add the option.

**Step 5** Click **Save**.

**Tip** If you add new option values or edit existing ones, be sure to save the policy object by clicking **Save**.

## CLI Commands

To view option values, use **policy** *name* **getOption** <*opt-name* | *id*> and **policy** *name* **listOptions**. To set option values, use **policy** *name* **setOption** <*opt-name* | *id*> *value* [**-blob**] [**-roundrobin**]. When you set an option value, the DHCP server replaces any existing value or creates a new one, as needed, for the given option name. When **-roundrobin** is enabled, instructs the DHCP server to return option data which contains more than one value in a different, rotated, order. A particular client will always get the same order, but different clients will get different "rotations" of the order of the multiple values configured for the option based on their client identifier. To unset option values, use **policy** *name* **unsetOption** <*opt-name* | *id*>.

# Adding Complex Values for Suboptions

If you are adding more complex option values such as for suboptions, use a parenthesized string format. The format requires that you:

- Enclose each option level (option, suboption, subsuboption) in parentheses.
- Separate multiple values with commas.
- Separate data fields for packed data (missing the suboption code or length) with semicolons.

For example, the *cablelabs-client-configuration* option (122) normally has 10 suboptions as well as some subsuboptions. This example shows the syntax to set the suboption 1, 2, 3, and 4 data values, and includes the two subsuboptions for suboption 3 and the three subsuboptions for suboption 4 (which are packed data and have no code numbers):

```
(primary-dhcp-server 1 10.1.1.10)
(secondary-dhcp-server 2 10.2.2.10)
(provisioning-server 3 (flag 0; provisioning-server server.example.com.))
(as-backoff-retry 4 (as-backoff-retry-initial-time-ms 10;
as-backoff-retry-max-time 10s; as-backoff-retry-count 100))
```

The suboption name (such as primary-dhcp-server) is optional. Hence, it is often safer to use just the code number and data value (or just the data value for packed data) to minimize typographical errors and parsing failures. The compacted (and preferred) version of the previous example that strips out the suboption names is:

```
(1 10.1.1.10) (2 10.2.2.10) (3 (0;server.example.com.)) (4 (10;10s;100))
```

Even if you use numerical code values, Cisco Prime Network Registrar always includes the equivalent names when it displays the suboptions (see Creating DHCP Option Definition Sets and Option Definitions, on page 11).

To include suboptions that include enterprise IDs (such as for option 125), use the following format, for example, when entering in the policy option value:

```
(enterprise-id 1((1 10.1.1.1) (2 10.2.2.2) (3 www.cisco.com)))
```

The parentheses surround the enterprise ID itself, the suboptions as a group, and each suboption.

## MAP-T and 4rd Options

The option value field can now specify encapsulated DHCPv6 options (i.e., top level options). So, options such as the Softwire MAP and 4rd options can be specified. For example:

```
nrcmd> policy softwire setv6option s46-cont-mapt "(s46-rule (flags 0; ea-len 12; prefix4-len
 24; ipv4-prefix 10.1.2.0; prefix6 1234::/64 (s46-portparams (offset 10; psid-len 10; psid
 43))))(s46-dmr 2345::/64)"
```

If you want the *s46-portparams* to be at the same "level" as the *s46-rule*, you should use:

```
nrcmd> policy softwire setv6option s46-cont-mapt "(s46-rule (flags 0; ea-len 12; prefix4-len
 24; ipv4-prefix 10.1.2.0; prefix6 1234::/64)) (s46-portparams (offset 10; psid-len 10;
psid 43))(s46-dmr 2345::/64)"
```

The syntax for sub-options and encapsulated options has changed to: *name id data*. Earlier, we used to require *id* and ignore *name* if it was present. Now, we no longer require the *id*. If *name* is present, it must be valid (it is no longer ignored). If *name* and *id* are present, the *name*'s id must match *id*. If you want "*name data*" where *data* is the option id, you must specify as "*name id data*", otherwise we will assume the *id* is the option id and not *data*.

For encapsulated options, there is no check made to assure that only the allowed options are specified; any option can be given.

# Creating and Editing Embedded Policies

An embedded policy is embedded for a DHCPv4 scope or scope template, DHCPv6 prefix or prefix template, client, or client-class. You can create or edit an embedded policy.

## Local Advanced Web and Regional UI

**Step 1** From the **Design** menu, choose one of the following that appear for DHCPv4 or DHCPv6 in the local web UI: **Scopes**, **Scope Templates**, **Clients**, **Client-Classes**, **Prefixes**, or **Links**. (The regional web UI can have the selections **Scope Templates**, **Client-Classes**, **Prefixes**, and **Links**.)

**Step 2** Click the name of the object on the left pane to open its Edit page.

**Step 3** Click **Create New Embedded Policy** or **Edit Existing Embedded Policy** under the Embedded Policy section of the page. This opens the Edit DHCP Embedded Policy page for the object.

**Step 4** Make changes to the values as needed, then click **Modify Embedded Policy**.

**Step 5** On the Edit page for the object, be sure to save the changes by clicking **Save**.

## CLI Commands

Use the embedded commands, such as **client-class-policy** *client-class-name* **set** *attribute*=*value*, where the command starts with the object name followed by -policy.

# Creating DHCP Option Definition Sets and Option Definitions

In Cisco Prime Network Registrar, you configure option values on policies for such things as lease times and router addresses. Numerous RFCs describe the formatting of DHCP option values, beginning with RFC 2132. Option definitions are used in the web UI and CLI to control formatting of option values in policies.

DHCPv6 options do not use DHCPv4 options; they are unique and separate. There are currently about 46 DHCPv6 options. Most of these options are the DHCPv6 protocol infrastructure options and are not user-definable. They use a 16-bit option code and 16-bit length (DHCPv4 uses only 8 bits for both of these). Configuring options and the behavior of configured options in policies are similar to those for DHCPv4. See Setting DHCPv6 Options, on page 20 for details about client processing as it relates to the policy hierarchy.

You can define option definitions separately for the DHCPv4 and DHCPv6 address spaces, as:

- **Standard (built-in) options** -Defined by the RFCs. In the web UI, these are in the **dhcp-config** and **dhcp6-config** definition sets. The CLI includes additional **dhcp-default** and **dhcp6-default** definition sets that are hidden, but accessible if you call for them specifically. (See Using Standard Option Definition Sets, on page 12.)
- **Custom options** -New or modified definitions in the supplied **dhcp-config** or **dhcp6-config** definition sets. Once you add or modify definitions in the web UI, they are added to the **dhcp-custom** or **dhcp6-custom** definition sets in the CLI. (See Creating Custom Option Definitions, on page 14.)
- **Vendor-specific options** -Defined in their own definition sets. The CableLabs definition sets (**dhcp-cablelabs-config** and **dhcp6-cablelabs-config**) are preconfigured in Cisco Prime Network Registrar. The CLI also includes **dhcp-cablelabs-default**, **dhcp6-cablelabs-default**, **dhcp-cablelabs-custom**, and **dhcp6-cablelabs-custom** definition sets. (See Using Standard Option Definition Sets, on page 12.)

# Related Topics

Using Standard Option Definition Sets, on page 12

Creating Custom Option Definitions, on page 14

Creating Vendor-Specific Option Definitions, on page 14

Option Definition Data Types and Repeat Counts, on page 20

Adding Suboption Definitions, on page 21

Importing and Exporting Option Definition Sets, on page 22

Pushing Option Definition Sets to Local Clusters, on page 23

Pulling Option Definition Sets from Replica Data, on page 23

Setting Option Values for Policies, on page 19

# Using Standard Option Definition Sets

Cisco Prime Network Registrar provides two standard, built-in option definition sets, **dhcp-config** and **dhcp6-config**, for DHCPv4 and DHCPv6 option definitions, respectively. You can create new options definitions in these sets or you can overwrite existing ones. New option definitions or ones that were overwritten are identified by an asterisk (*). You can delete these definitions and there is no deletion confirmation given. However, saving the set after deleting an overwritten definition causes the original definition to reappear in the set.

**Caution** Arbitrarily modifying the standard definitions (or adding suboption definitions) can adversely affect configurations.

## Local Advanced and Regional Web UI

**Step 1** From the **Design** menu, choose **Options** under the **DHCPv4** or **DHCPv6** submenu to open the List/Add DHCP Option Definition Sets page. (DHCP option definition is not available in Basic mode.)

**Step 2** Click the **dhcp-config** (DHCPv4) or **dhcp6-config** (DHCPv6) link to open the Edit DHCP Option Definition Set page, then click the **Option Definitions** tab. View the predefined definitions on the List/Add DHCP Option Definitions page. These are the definitions that control the formatting of the option values you add to policies. If there are suboption definitions, you can expand to show them.

**Step 3** To add a definition, click the **Add Option Definition** icon on the Edit DHCP Option Definition page. Give the option an number, name, description, type, and repeat count (whether more than one instance of the option is allowed or required). (For details on the data types and repeat count values, see Option Definition Data Types and Repeat Counts, on page 20.)

**Note** You cannot add an option definition for an option number or name that already exists. However, you can modify any option definition that appears as a hyperlink on the page.

**Step 4** Click **Add Option Definition**. Then, on the List/Add DHCP Option Definition Sets page, click **Save**.

**Step 5** Click the **Cancel** button if you want to revert to the original definitions in that standard set.

**Step 6** In the regional web UI, you can also pull replica definition sets and push definition sets to local clusters. (See Pulling Option Definition Sets from Replica Data, on page 23 and Pushing Option Definition Sets to Local Clusters, on page 23.)

## CLI Commands

To view the entire list of standard DHCP option definitions, use **option-set dhcp-config** [**show**] or **option-set dhcp6-config** [**show**], or **option** {*id* | *name*} *option-set* **show** to view a specific definition. For example:

```
nrcmd> option-set dhcp-config
nrcmd> option subnet-mask dhcp-config show
```

To add a definition to a set, use **option** *id option-set* **create** *option-name type* [*attribute*=*value*]. You cannot add a definition for an option ID (number) or name that already exists. For example, to add option number 222 with the name example-option in the dhcp-config option set, with a string type, use:

```
nrcmd> option 222 dhcp-config create example-option AT_STRING
```

To get a particular option attribute value, use **option** {*id* | *name*} *option-set* **get** *attribute*. To modify an option attribute, use **option** {*id* | *name*} *option-set* **set** *attribute*=*value*. You can also unset an option attribute.

When connected to a regional cluster, you can use the following pull, push, and reclaim commands. For push and reclaim, a list of clusters or "all" may be specified.

- **option-set** < *name* | **all** > **pull** < **ensure** | **replace** | **exact** > *cluster-list* [**-report-only** | **-report**]
- **option-set** < *name* | **all** > **push** < **ensure** | **replace** | **exact** > *cluster-list* [**-report-only** | **-report**]
- **option-set** *name* **reclaim** *cluster-list* [**-report-only** | **-report**]

# Creating Custom Option Definitions

You can create custom option definitions in the standard sets. Click the **dhcp-config** or **dhcp6-config** set on the List/Add DHCP Option Definition Sets page. Then proceed with **Step 3** in Using Standard Option Definition Sets, on page 12.

# Creating Vendor-Specific Option Definitions

You can send vendor-specific option data to DHCP clients that request them.

**Note** There are several option codes set aside for vendor-specific options, so that you must explicitly specify the option code number for which you are creating a vendor-specific option definition.

In Cisco Prime Network Registrar, you can create vendor-specific option definitions in the web UI, or in the CLI by using **option** *id option-set-name* **create**. (For details on the option data types, see Option Definition Data Types and Repeat Counts, on page 20.)

Vendor-specific options are sent in the following DHCP options:

- ***vendor-encapsulated-options (43)***—Set this to a binary data type, then add the vendor-specific suboption definitions. (The data type of the parent option definition is a placeholder only. The suboption definitions define the valid option value formatting.)
- ***v-i-vendor-info (125) or vendor-options (17) for DHCPv6***—Set this to a vendor-opts data type, then add the vendor-specific suboption definitions.

You can create vendor-specific option definitions for DHCPv4 options 43 and 125, and DHCPv6 option 17. You add the vendor-specific option definitions into a vendor option definition set that you create.

**Caution** Changing option definition properties, or deleting the option definition altogether, can have unexpected side effects on policies. If you delete a custom option definition, also check for the policies that include an option value. Changing an option definition changes the way that they are displayed, not what is stored, so that you do not need to modify the policy value unless you want the policy to return a differently formatted option value. Some option types are very similar, and changing between them can have side effects. For example, strings and DNS names are both entered as string values in the user interfaces, but the formatted option values are quite different.

**Note** Cisco Prime Network Registrar 7.0 preconfigures separate CableLabs (enterprise ID 4491) option definitions in the **dhcp-cablelabs-config** and **dhcp6-cablelabs-config** vendor-specific option definition sets.

## Local Advanced and Regional Web UI

**Step 1** From the **Design** menu, choose **Options** under the **DHCPv4** or **DHCPv6** submenu to open the List/Add DHCP Option Definition Sets page. View the existing DHCPv4 or DHCPv6 options.

**Step 2** Click the **Add Options** icon in the Options pane to open the Add OptionDefinitionSet dialog box.

**Step 3** Enter a name for the option definition set, then choose DHCPv4 or DHCPv6 from the DHCP Type drop-down list.

If you are creating vendor-specific option definitions using:

- Option 43, enter a value in the Vendor Option String field. (See the subsequent section for a sample procedure on creating a vendor option set and vendor option values for option 43.)
- Option 125 for DHCPv4 or option 17 for DHCPv6, enter a valid Enterprise Option Enterprise ID value.

**Step 4** Click **Add OptionDefinitionSet**.

**Step 5** Click the added option definition set name on the left pane.

**Step 6** On the Edit DHCP Option Definition Set page, click the **Option Definitions** tab. Any existing option definitions will appear on this page (new or modified standard definitions are marked with an asterisk).

**Step 7** Click **Add Option Definition** icon. Enter the ID number of the option definition, along with its name and a description. The ID must be 43, 125, or 17 (for DHCPv6) for the client to recognize a vendor-specific option definition. The option name does not need to match the one specified in the RFC and can be of your own creation.

**Step 8** Choose a data type and repeat count (or enter an absolute repeat count in the next field). The data type must be:

- Binary (AT_BLOB) for option 43.
- Vendor-opts (AT_VENDOR_OPTS) for option 125 (for DHCPv4) and option 17 (for DHCPv6).

(For details on the data type and repeat count values, see Option Definition Data Types and Repeat Counts, on page 20.)

**Step 9** Click **Add Option Definition**. Then, on the List DHCP Option Definitions page, click **Save**.

---

Using the Local Advanced web UI to create vendor option set and vendor option values for option 43:

---

**Step 1** From the **Design** menu, choose **Options** under the **DHCPv4** or **DHCPv6** submenu to open the List/Add DHCP Option Definition Sets page.

**Step 2** Click the **Add Options** icon in the **Options** pane to open the Add OptionDefinitionSet dialog box.

**Step 3** Enter values for the following attributes:

- **Name**—Name of the option definition set; for example, AP1130.
- **DHCP Type**—Byte size of the type identifiers for all children in this set. You must choose DHCP v4 from the drop-down list.
- **Vendor Option String**—Exact vendor class identifier string from option-60 that the DHCP client device vendor provides. For example, Cisco AP c1130.

**Step 4** Click **Add OptionDefinitionSet**.

The List/Add DHCP Option Definition Sets page appears.

**Step 5** Click AP1130, the name of the option definition set that appears.

The Edit DHCP Option Definition Set AP1130 page appears.

**Step 6** Click the **Option Definitions** tab and then click the **Add Option Definition** icon.

**Step 7** Enter the values for the following attributes:

- **Number**—Number of the option code. You must enter 43.

- **Name**—Number of the option code. You must enter 43.
- **Type**—Name of this attribute. For example, ap1130-option-43.

**Step 8** Click **Add Option Definition**.

Note that clicking this button does not save the changes that you make to the option definition set. It only lists the option definition set on the List DHCP Option Definitions page.

**Step 9** In the Option Definitions tab, click the name of the new option definition (ap1130-option-43), then click **Add Sub-Option Definition**.

**Step 10** Enter values for the following attributes:

- **Number**—The option code for this suboption. For this example, you must enter 241.
- **Name**—Name of this attribute. For example, "ap1130-suboption-241".
- **Type**—Datatype for the suboption value. For this example, you must choose IP Address from the drop-down list.
- **Repeat**—The repeat count for this type. For this example, you must choose 1+ from the drop-down list.

**Step 11** Click **Add Option Definition**, then click **Save**.

**Step 12** From the **Design** menu, choose **Policies** under the **DHCP Settings** submenu to open the List/Add DHCP Policies page.

**Step 13** Choose the policy for which to set this option; or, add a new policy in the Advanced mode.

Depending on your selection, the Edit DHCP Policy *policy_name* or the Add DHCP Policy page appears.

**Step 14** From the DHCPv4 Vendor Options drop-down list, choose the name of the option definition set (AP1130), and click **Select**.

**Step 15** Choose the option definition from the Name drop-down list (“ap1130-option-43”) and enter the value in the Value field. For example:

```
(241 3.3.3.3,4.4.4.4)
```

**Step 16** Click **Add Option**, then click **Save**.

**Step 17** Reload the DHCP server.

## Example: Creating Vendor Option Set for Cisco AP Devices

You can create a vendor option set and vendor option values from the CLI for Cisco Access Point (AP) devices, SunRay devices, and Cisco 79xx IPPhones using the sample procedures described in this section.

Using option 43 for Lightweight Access Point Protocol (LWAPP) APs requires vendor option 43 if you are using Cisco Prime Network Registrar as the DHCP server. This example is specific to the Cisco Aironet 1130 series. You can modify the example to configure option 43 for other vendor options, such as Cisco Aironet 1200 series and Cisco Aironet 1240 series.

**Step 1** Create a .txt file with the following content:

```
#
# Version: 1
# 6.2+ Option-set example for Option 43 with suboptions for Cisco APs
#
# NOTE: Need to edit vendor option string to Exact match AP Model string in Option-60.
```

```
#
# For compatibility with pre-6.2 vendor options ensure that
# name=vendor-option-string. (Not True in this test example.)
# ======================================================================
{
( id-range = 1 )
( vendor-option-string = Cisco AP c1130 )
( name = APtest )
( children = [
{
( id = 43 )
( name = pxe-sample )
( desc = )
( base-type = AT_BLOB )
( children = [
{
( id = 241 )
( name = controller )
( desc = ap controller )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
} ]
)
} ]
)
}
```

**Step 2** Save the file as *OptionSetCiscoAP.txt* at the following location:

- Windows—\Program Files\Network Registrar\Local\bin
- Linux—/opt/nwreg2/local/usrbin

**Step 3** Import the OptionSetCiscoAP.txt file from the CLI using the import option-set file command. For example:

```
nrcmd> import option-set OptionSetCiscoAP.txt
```

(For information on importing option definition sets, see Importing and Exporting Option Definition Sets, on page 22.)

**Step 4** Set the vendor-specific option data on a policy using the **policy** *name* **setVendorOption** <*opt-name* | *id*> *opt-set-name value* [**-blob**] command.

For example, to set vendor option 43 data for the optionset APtest with values (241 3.3.3.3,4.4.4.4), on an existing policy with the name test, use:

```
nrcmd> policy test setVendorOption 43 APtest "(241 3.3.3.3,4.4.4.4)"
nrcmd> save
```

**Step 5** Reload the DHCP server.

```
nrcmd> dhcp reload
```

## Example: Creating Vendor Option Set for SunRay Devices

Use this sample procedure to create vendor option set with multiple suboptions for SunRay Devices:

**Step 1** Create a .txt file with the following content:

```
#
# Option Definition Set Export/Import Utility
# Version: 1
# 6.2 Option-set example for Option 43 with suboptions for Sun SunRay.
#
# NOTE: Need to edit vendor option string to match Option-60
#
# For compatibility with pre-6.2 vendor options ensure that
# name=vendor-option-string.
# ======================================================================
{
( id-range = 1 )
( vendor-option-string = sunray )
( name = sunray )
( children = [
{
( id = 43 )
( name = option43 )
( desc = )
( base-type = AT_BLOB )
( children = [
{
( id = 21 )
( name = AuthSrvr )
( desc = AuthSrvr )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
}]
{
( id = 35 )
( name = AltAuth )
( desc = AltAuth )
( base-type = AT_IPADDR )
( repeat = ONE_OR_MORE )
}
{
( id = 36 )
( name = BarrierLevel )
( desc = BarrierLevel )
( base-type = AT_SHORT )
}
]
)
} ]
)
}
```

**Step 2** Save the file as *OptionSetSunRay.txt* at the following location:

- Windows—\Program Files\Network Registrar\Local\bin
- Linux—/opt/nwreg2/local/usrbin

**Step 3** Import the OptionSetSunRay.txt file from the CLI using the import option-set file command. For example:

```
nrcmd> import option-set OptionSetSunRay.txt
```

(For information on importing option definition sets, see Importing and Exporting Option Definition Sets, on page 22.)

**Step 4** Set the vendor-specific option data on a policy using the **policy** *name* **setVendorOption** <*opt-name* | *id*> *opt-set-name value* [**-blob**] command.

For example, to set vendor option 43 data for the optionset APtest with values (241 3.3.3.3,4.4.4.4), on an existing policy with the name test, use:

```
nrcmd> policy test setVendorOption 43 APtest "(241 3.3.3.3,4.4.4.4)"
nrcmd> save
```

**Step 5** Reload the DHCP server.

```
nrcmd> dhcp reload
```

## Example: Creating Option Set for Cisco 79xx IPPhones

Use this sample procedure to create option set for Cisco 79xx IPPhones:

**Step 1** Define the option.

```
nrcmd> option 150 dhcp-custom create voip-tftp-server AT_IPADDR desc="VOIP Option-150 Server”
repeat=ONE_OR_MORE
```

**Step 2** Display the configured option.

```
nrcmd> option dhcp-config list
```

**Step 3** Set policy, by using **policy default setOption voip-tftp-server ip-address**. For example:

```
nrcmd> policy default setOption voip-tftp-server 192.168.1.254
```

**Step 4** Confirm the policy setting.

```
nrcmd> policy default getOption voip-tftp-server
```

**Step 5** Reload the DHCP server.

```
nrcmd> dhcp reload
```

# Setting Option Values for Policies

You enter option values on a policy. The option definitions in your server configuration control the format and values that you enter.

## Local Advanced and Regional Web UI

On the List/Add DHCP Policies page, click a policy to edit it. (Note that you cannot set options for policies in Basic mode.) On the Edit DHCP Policy page:

- To enter a standard DHCPv4 or DHCPv6 option value for a policy, choose it from the DHCPv4 Options or DHCPv6 Options drop-down list, then set a value for the option. Click **Add Option**.
- To enter a vendor-specific DHCPv4 or DHCPv6 option value for a policy, choose an option definition set in the DHCPv4 Vendor Options or DHCPv6 Vendor Options drop-down list, then click **Select**. The page changes to show the drop-down list that includes the option; choose it, then click **Add Option**.

Note that you can also edit policy attributes on this page. Be sure to click **Modify Policy**.

To edit a configured policy option, click the name of the configured option on the Edit DHCP Policy page to open the Edit DHCP Policy Option page. Enter a new value, then click **Modify Option**.

### CLI Commands

Use one of these commands:

```
nrcmd> policy name setOption {opt-name | id} value [-blob] [-roundrobin]
nrcmd> policy name setV6Option {opt-name | id}[.instance] value [-blob] [-roundrobin]
nrcmd> policy name addV6Option {opt-name | id}[.instance] value [-blob] [-roundrobin]
nrcmd> policy name setVendorOption {opt-name | id} opt-set-name value [-blob]
nrcmd> policy name setV6VendorOption {opt-name | id} opt-set-name value [-blob]
```

To list the options in the policy, use one of these commands:

```
nrcmd> policy name listOptions
nrcmd> policy name listV6Options
nrcmd> policy name listVendorOptions
nrcmd> policy name listV6VendorOptions
```

To add suboption values, see Adding Complex Values for Suboptions, on page 10.

## Setting DHCPv6 Options

Set DHCPv6 options and vendor options when you create or edit policies (embedded or named) for prefixes. (See DHCPv6 Policy Hierarchy, on page 5 for special handling of the *v6-options* and *v6-vendor-options* policy attributes when used in an embedded or named policy on a prefix.)

### Local Advanced Web UI

The DHCPv6 options coexist along with the DHCPv4 options on the List/Add DHCP Policies or Edit DHCP Policy page. Note that the vendor options appear only if you create these options (see Creating DHCP Option Definition Sets and Option Definitions, on page 11).

You can select the options from the drop-down lists. If option descriptions exist, they appear under the Name and Number headings, which you can click to sort the entries.

### CLI Commands

Use **policy** *name* **setV6Option** {*opt-name* | *id*}[.*instance*] *value* [**-blob**] [**-roundrobin**] or **policy** *name* **setV6VendorOption** {*opt-name* | *id*} *opt-set-name value* [**-blob**]. When **-roundrobin** is enabled, instructs the DHCP server to return option data which contains more than one value in a different, rotated, order. A particular client will always get the same order, but different clients will get different "rotations" of the order of the multiple values configured for the option based on their client identifier. The option settings require an option name (or ID) and a value. For example:

```
nrcmd> policy dhcpv6-policy setV6Option dns-servers 2222::1,2222::2

nrcmd> policy foo setV6VendorOption 17 dhcp6-cablelabs-config "(32 2222::3,2222::4)"
```

## Option Definition Data Types and Repeat Counts

The data type values that you can use appear in the following table.

*Table 2: Option Definition Data Types*

| | | | |
|---|---|---|---|
| AT_INT8 unsigned 8-bit | AT_SHORT unsigned 16-bit | AT_INT unsigned 32-bit | AT_STRING string |
| AT_SINT8 signed 8-bit | AT_SSHORT signed 16-bit | AT_SINT signed 32-bit | AT_NSTRING string (no termination) |
| AT_DNSNAME DNS name | AT_SHRTI unsigned 16-bit (Intel) | AT_INTI unsigned 32-bit (Intel) | AT_BLOB binary |
| AT_RDNSNAME relative DNS name | AT_SSHRTI signed 16-bit (Intel) | AT_SINTI signed 32-bit (Intel) | AT_DATE date |
| AT_VENDOR-CLASS vendor-class | AT_IPADDR IP address | AT_BOOL boolean | AT_TIME unsigned time |
| AT_VENDOR_NOLEN vendor-nolen | AT_IP6ADDR IPv6 address | AT_MACADDR MAC address | AT_STIME signed time |
| AT_VENDOR_OPTS vendor-opts | AT_ZEROSIZE zero size | AT_VPREFIX IPv6 variable-legth prefix | |

You can view these types in the CLI by using **option listtypes**.

To set the repeat count, set the *repeat-count* attribute to one of the following, or enter an absolute number:

- **ZERO_OR_MORE**—**0+** in the web UI
- **ONE_OR_MORE**—**1+** in the web UI
- **EVEN_NUMBER**—**2n** in the web UI

In the CLI, for example, use:

```
nrcmd> option 200 ex-opt-def-set set repeat-count=ZERO_OR_MORE

nrcmd> save
```

# Adding Suboption Definitions

You can set a suboption definition for the option definition by clicking **Add Suboption Definition** on the Edit DHCP Option Definition page. This opens the Add DHCP Option Definition page, where you can add the same values as for an option definition. The suboption definition you create is associated with its parent option (or parent suboption) definition. You can define up to six option and suboption levels.

**Note** You can add suboption definitions by using the web UI only. You currently cannot do so by using the CLI.

Suboption definition formats can be packed or type/length/value (TLV):

- **Packed**—A suboption with a zero ID value and an implicit data type. The option value is the only data in the packet. DHCPv6 options are virtually all defined with packed data. There are no markers for type or length and the layout of the data is inherent in the option definition. You cannot have further suboption definitions for packed suboptions.

- **TLV**—A suboption with a value of 1 through 255 (or 65535) that includes a type, length, and value. The data in the packet has the type and length preceding the value.

In most cases, you will not be mixing packed with TLV suboptions for the same option.

**Note** The DHCP server does not support suboption 0 definitions (for the DHCPv4 vendor-encapsulated-options (43) and v-i-vendor-opts (125) options, and DHCPv6 vendor-opts (17) option). Suboption with a zero ID value is used by the DHCP server to specify packed data as mentioned above.

To enter suboption values when editing policies, see Adding Complex Values for Suboptions, on page 10.

# Option Definition Set

## Importing and Exporting Option Definition Sets

Importing and exporting option definition sets is a way to copy them between servers. In the CLI, you can import and export option sets by using **import option-set** *file* and **export option-set** *name file*.

For example, to import an option set for Preboot Execution Environment (PXE) clients, modify and import a sample file located in the /examples/dhcp directory:

```
nrcmd> import option-set /examples/dhcp/OptionSetPXE.txt
```

**Caution** Do not export the built-in option definition sets (such as dhcp-config and dhcp-cablelabs-config) and then reimport them. Reimporting an edited option definition set without TAC assistance can cause the server to fail.

Some of the guidelines for the file format include:

- The version string in the file must match the version for the import utility.
- The utility imports just the first option definition set found in the file.
- Delimit objects using curly brackets (**{ }**), attributes using parentheses (**( )**), and lists of objects in attributes using square brackets (**[ ]**). Delimit string value attributes using quotes (" ").

Using some care, you can also edit the text file to make minor modifications to an option definition set. Cisco Prime Network Registrar provides two sample option definition set text files in the examples/dhcp directory, OptionSetJumpStart.txt and OptionSetPXE.txt:

- **OptionSetJumpStart.txt**—Edit the vendor-option-string to match the dhcp-class-identifier (option 60) that your JumpStart clients are sending.
- **OptionSetPXE.txt**—Edit the vendor-option-string to match the dhcp-class-identifier (option 60) that your Pre-boot Execution Environment (PXE) clients are sending.

## Pushing Option Definition Sets to Local Clusters

You can push option definition sets you create from the regional cluster to any of the local clusters. If you want to push a specific option definition set to a cluster, click **Push Option Definition** sets on the List/Add DHCP Option Definition Sets page, which opens the Push DHCP Option Definition Set to Local Clusters page.

This page identifies the data to push, how to synchronize it with the local cluster, and the cluster or clusters to which to push it. The data synchronization modes are:

- **Ensure** (preset value)—Ensures that the local cluster has new data without affecting any existing data.
- **Replace**—Replaces data without affecting other objects unique to the local cluster.
- **Exact**—Available for "push all" operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the local cluster.

Choose the destination cluster or clusters in the Available field and move it or them to the Selected field.

**Tip** The synchronization mode and cluster choice settings are persistent for the duration of the current login session, so that they are in effect each time you access this page, unless you change them.

After making these choices, click **Push Data to Clusters**. This opens the View Push DHCP Option Definition Set Data Report page.

## Pulling Option Definition Sets from Replica Data

You may choose to pull option definition sets from the replica data of the local clusters instead of explicitly creating them. (You may first want to update the option definition set replica data by clicking the **Replicate** icon next to the cluster name.) To pull the option definition sets in the web UI, click **Pull Replica Option Definition Sets** to open the Select Replica DHCP Option Definition Set Data to Pull page.

This page shows a tree view of the regional server replica data for the local clusters' option definition sets. The tree has two levels, one for the local clusters and one for the scope templates in each cluster. You can pull individual option definition sets from the clusters, or you can pull all of their option definition sets. To pull individual ones, expand the tree for the cluster, then click **Pull Option Definition Set** next to its name. To pull all the ones from a cluster, click **Pull All Option Definition Sets from Cluster**. To pull the option definition sets, you must also choose a synchronization mode:

- **Ensure**—Ensures that the regional cluster has new data without affecting any existing data.
- **Replace** (preset value)—Replaces data without affecting other objects unique to the regional cluster.
- **Exact**—Available for "pull all" operations only. Use this with caution, because it overwrites the data and deletes any other objects unique to the regional cluster.