Managing Caching DNS Server Properties

This chapter explains how to set the Caching DNS server parameters. Before you proceed with the tasks in this chapter, see Chapter 14, "Introduction to the Domain Name System" which explains the basics of DNS.

Related Topics

Managing DNS Caching Servers

You can view its health, statistics, and logs; start, stop, and reload it; run certain commands (see the "Running DNS Caching Server Commands" section on page 19-1); and edit the server attributes.

To view the server status and health, or stop, start, and reload the server, in the local cluster web UI, choose **CDNS Server** from the **Deploy > DNS** menu to open the Manage DNS Caching Server page.

Related Topics

Running DNS Caching Server Commands

Access the commands by using the Commands button. Clicking the Commands button opens the CDNS Commands dialog box in the local web UI. Each command has its own Run icon (click it, then close the dialog box):

- **Flush the CDNS cache**— This command allows you to flush either all RRs or RRs for a particular zone from the in-memory cache. See the "Flushing CDNS Cache" section on page 19-10
- **Flush Resource Record**— This command that lets you specify an RR name and optionally a type to remove from the in-memory cache.

Note To remove all the entries from the in-memory cache, you need to reload the CDNS server.

Note If you find a server error, investigate the server log file for a configuration error, correct the error, return to this page, and refresh the page.

Configuring CDNS Server Network Interfaces

You can configure the network interfaces for the CDNS server from the Manage Servers page in the local web UI.

Local Advanced Web UI

Step 1 From the **Operate** menu, choose **Manage Servers** under the **Servers** submenu.

Step 2 Select **Local CDNS Server** from the Manage Servers pane.

Step 3 Click the **Network Interfaces** tab to view the available network interfaces that you can configure for the server. By default, the server uses all of them.

Step 4 To configure an interface, click the **Configure** 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 5 Click the name of the configured interface to edit the configured interfaces, where you can change the address, direction and port of the interface.

Step 6 Click **Modify Interface** when you are done editing, then click **Go to Server Interfaces** to return to the Network Interfaces page.

Defining Forwarders

You can specify a domain for which forwarding should occur. The forwarder definition is by a list of names of servers or a list of IP addresses with an optional port number, or both.

Note You can specify IPv4 and/or IPv6 addresses and for the changes to take effect, you must reload the CDNS server.

Tip To force a caching DNS server to only talk to a forwarder, define a forwarder for the DNS root (.).

Local Basic or Advanced Web UI

To define a forwarder:

Step 1 From the **Design** menu, choose **Forwarders** under the **Cache DNS** submenu. This opens the List/Add Forwarders page.

Step 2 Click the **Add Forwarders** icon in the Forwarders pane to open the Add DnsForwarder dialog box.

Step 3 Enter the forwarder name and click **Add DnsForwarder**.

Step 4 In the Edit Forwarders page, enter the hostname, and click **Add Host** and enter the IP address for the forwarder then click **Add Address**.

Step 5 Click **Save**.

CLI Commands

Use the following cdns commands to:

- Specify the address (or space-separated addresses) of nameservers to use as forwarders, use **cdns addForwarder**.
- List the current forwarders, use **cdns listForwarders**.
- Edit your forwarder list, you must remove any offending forwarder and reenter it.
- Remove a forwarder or list of forwarders, use **cdns removeForwarder**.

Note For any change to the forwarders to take effect, you should restart the CDNS server.

Using Exceptions

If you do not want the CDNS server to use the standard resolution method to query the root nameserver for certain domains, use exceptions. This bypasses the root nameservers and targets a specific server (or list of servers) to handle name resolution.

Let us say that example.com has four subsidiaries: Red, Blue, Yellow, and Green. Each has its own domain under the .com domain. When users at Red want to access resources at Blue, their CDNS server follows delegations starting at the root nameservers.

These queries cause unnecessary traffic, and in some cases fail because internal resources are often barred from external queries or sites that use unreachable private networks without unique addresses.

Exceptions solve this problem. The Red administrator can list all the other example.com domains that users might want to reach and at least one corresponding nameserver. When a Red user wants to reach a Blue server, the Red server queries the Blue server instead following delegations from the root servers down.

To enable resolution exceptions, simply create an exception for the domain listing the IP address(es) and/or hostname(s) of the authoritative nameserver(s).

Note Exceptions can contain both IPv4 and/or IPv6 addresses and require a CDNS server reload to take effect.

Local Basic or Advanced Web UI

Step 1 From the **Design** menu, choose **Exceptions** under the **Cache DNS** submenu. This opens the List/Add Exceptions page.

Step 2 Click the **Add Exceptions** icon in the Exceptions pane to open the Add DnsException dialog box.

Step 3 In the name field, enter the domain or zone for which an exception is wanted and click **Add DnsException**.

Step 4 In the Edit Exceptions page, enter the hostname in the DNS Name field and click **Add Host**. To address, enter the IP address in the IP Address field and click **Add Address**.

Step 5 If the prime attribute is on, CDNS queries the zone for the currently published name servers and use those. This is similar to how the server treats root hints.

Step 6 Click **Save**.

To delete an exception list, select the exception in the Exceptions pane and click the **Delete** icon. To add or remove name servers to an exception, click the name of the exception in the List/Add Exceptions page to open the Edit Exceptions page.

CLI Commands

Use the exception commands only if you do not want your DNS Caching server to use the standard name resolution for querying root name servers for names outside the domain. Network Registrar sends non-recursive queries to these servers.

Use the following cdns commands to:

- Add the resolution exception domains and the IP addresses of servers, separated by spaces, use **cdns addException** *domain* [*prime=on\off*] [*views=on\off*] *addr*. The addresses can be IPv4 or IPv6 with an optional port number (i.e. <addr>[@<port>]) or the name of a server (it must be possible to resolve the server name before it is used). Use this command only if you do not want your DNS Caching server to use the standard name resolution for a zone.
- List the domains that are configured to have exceptional resolution of their names, use **cdns listExceptions.**
- Remove an entry for exceptional resolution of addresses within a domain, use **cdns removeException**. You can remove an individual server by specifying it, or the exception itself by just specifying its name.
- Replace an exception, you must first remove the current exception and then add a new one.

For any change to resolution exceptions to take effect, you must restart the CDNS server.

Managing DNS64

DNS64 with NAT64 provides access to the IPv4 Internet and servers for hosts that have only IPv6 addresses. DNS64 synthesizes AAAA records from A records, when a IPv6 client queries for AAAA records, but none are found. It also handles reverse queries for the NAT64 prefix(es).

Local Advanced Web UI

Step 1 From the **Design** menu, choose **DNS64** under the **Cache DNS** submenu. This opens the Manage DNS64 page.

Step 2 Check the **true** option if you want to enable the DNS64 processing.

Step 3 If needed, add the IPv6 prefix to use for synthesizing AAAA records in the **Prefix** field. The prefix length must be 32, 40, 48, 56, 64, or 96, and bits 64-71 of the prefix must be zero.

In the **Expert** mode, you have the following extra options:

- to specify the IPv6 Suffix to use for synthesizing AAAA records in the **Suffix** field. The suffix is ignored if the dns64 prefix is 96 bits long.
- to set to **true** the *synthesize-all* attribute which forces DNS64 to always synthesize AAAA records from A records when they are requested.

Step 4 Click **Save** to save your settings.

CLI Commands

To create DNS64 in the DNS Caching server, use **cdns64 create**. To enable DNS64, use **cdns64 enable dns64** (see the cdns64 command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

Managing DNSSEC

DNSSEC enables the server to determine the security status of all Resource Records that are retrieved. You can manage DNSSEC only in the Advanced mode. DNSSEC requires a root trust anchor to establish trust for the DNS root servers. The initial DNSSEC root trust anchor, root.anchor, is stored in the .../data/cdns directory and is the default value of the *auto-trust-anchor-file* attribute. Additional trust anchors may be added by adding them to the .../data/cdns directory and to the *auto-trust-anchor-file* if the zone supports automated updates according to RFC 5011 or the *trust-anchor-file* attribute if not. The **cdnssec** command controls and configures DNSSEC processing in the Cisco Prime Network Registrar DNS Caching server.

Local Basic or Advanced Web UI

Step 1 From the **Design** menu, choose **DNSSEC** under the **Security** submenu to open the Manage DNSSEC page.

Step 2 Enable DNSSEC validation by selecting the **enabled** option.

Step 3 The page displays all the DNSSEC attributes. Modify the attributes as per your requirements.

Step 4 Click **Save** to save your settings.

CLI Commands

To create DNSSEC in the DNS Caching server, use **cdnssec create**. To enable DNS64, use **cdns64 enable dnssec** (see the cdnssec command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

Setting DNS Caching Server Properties

You can set properties for the Caching DNS server. These include:

- **General server properties**—See the "Setting General CDNS Server Properties" section on page 19-6.

- **Log Settings—**See the "Specifying Log Settings" section on page 19-6.
- **Activity Summary Settings**—See the "Specifying Activity Summary Settings" section on page 19-7
- **Caching Settings** —See the "Specifying Caching Settings" section on page 19-7.
- **Cache TTLs**—See the "Setting Cache TTLs" section on page 19-7.
- **Root name servers**—See the "Defining Root Nameservers" section on page 19-8.
- **UDP Ports**—See the "Dynamic Allocation of UDP Ports" section on page 19-8

Setting General CDNS Server Properties

You can view CDNS general server properties, such as log settings, basic cache settings, SNMP traps, and root nameservers.

The following subsections describe some of the most common property settings. They are listed in the "Setting DNS Caching Server Properties" section on page 19-5.

Local Basic or Advanced Web UI

Step 1 To access the server properties, choose **CDNS Server** from the **Deploy > DNS** submenu to open the Manage DNS Caching Server page.

Step 2 Select **Local CDNS Server** from the CDNS Server pane, to open the Edit Local CDNS Server page. The page displays all the CDNS server attributes.

Step 3 Click **Save** to save the CDNS server attribute modifications.

CLI Commands

Use **cdns show** to display the CDNS server properties (see the **cdns** command in the CLIGuide.html file in the /docs directory for syntax and attribute descriptions).

Specifying Log Settings

This setting determines which detailed events the Caching DNS server logs, as set using a bit mask. Logging these additional details can help analyze a problem. Leaving detailed logging enabled for a long period, however, can fill the log files and cause the loss of important information.

The possible options are:

- **config**—Controls logging pertaining to server configuration and server de-initialization (unconfiguration).
- **server-ops**—Controls high level logging of server operations.
- **server-detailed-ops**—Controls detailed logging of server operations.
- **scp**—Controls logging pertaining to SCP message processing.
- **activity-summary**—This causes a summary message to appear at an interval specified by activity-summary-interval. The summary provides detailed statistics about the servers operation.
- **query**—Causes logging of all DNS queries to the server.

Specifying Activity Summary Settings

Note To specify the activity summary settings, you have to check *activity-summary* under the Log Settings.

You can specify the interval at which to log activity-summary information using the Statistics Interval (*activity-summary-interval*) attribute.

The Caching DNS server logs sample and/or total statistics based on the option you check for the attribute Statistics Type (*activity-summary-type*).

The option checked for the attribute Statistics Settings (*activity-summary-settings*) determines the category of statistics that is logged as part of activity summary. The possible settings are:

- query—Logs statistics related to incoming queries.
- query-type—Logs statistics on the RR types that are being queried.
- cache—Logs statistics on the RR cache.
- resol-queue—Logs statistics on the resolution queue.
- responses—Logs statistics about query responses.
- memory—Logs statistics on memory usage.
- redirect— Logs statistics on redirect usage.

Specifying Caching Settings

To set the cache TTLs, see the "Setting Cache TTLs" section on page 19-7.

Use the *Prefetch* attribute to set whether message cache elements should be prefetched before they expire to keep the cache up to date. Turning it **on** gives about 10 percent more traffic and load on the machine, but popular items do not expire from the cache.

When prefetch is enabled, records are assigned a prefetch time that is within 10 percent of the expiration time. As the server processes client queries and looks up the records, it checks the prefetch time. Once the record is within 10 percent of its expiration, the server will issue a query for the record in order to keep it from expiring.

Setting Cache TTLs

TTL is the amount of time that any nameserver is allowed to cache data learned from other nameservers. Each record added to the cache arrives with some TTL value. When the TTL period expires, the server must discard the cached data and get new data from the authoritative nameservers the next time it sends a query. TTL attributes, *cache-min-ttl* and *cache-max-ttl* defines the minimum and the maximum time Cisco Prime Network Registrar retains the cached information. These parameters limit the lifetime of records in the cache whose TTL values are very large.

Local Basic and Advanced Web UI

Step 1 On the Edit Local CDNS Server tab, in A-Z view, you can find:

- the Maximum Cache TTL (*cache-max-ttl*) attribute, set it to the desired value (the preset value is 24 hours)

- the Min Cache TTL (*cache-min-ttl*) attribute, set it to the desired value (the preset value is 0)

Step 2 Click **Save** to save the changes.

CLI Commands

Use:

- c**dns set cache-max-ttl** to set the Maximum Cache TTL.
- **cdns set cache-min-ttl** to set the Minimum Cache TTL.

Defining Root Nameservers

Root nameservers know the addresses of the authoritative nameservers for all the top-level domains. When you first start a newly installed Cisco Prime Network Registrar DNS server, it uses a set of preconfigured root servers, called root hints, as authorities to ask for the current root nameservers.

When Cisco Prime Network Registrar gets a response to a root server query, it caches it and refers to the root hint list. When the cache expires, the server repeats the process. The time to live (TTL) on the official root server records is currently six days, so Cisco Prime Network Registrar requeries every six days, unless you specify a lower maximum cache TTL value (see the "Setting Cache TTLs" section on page 19-7).

Because the configured servers are only hints, they do not need to be a complete set. You should periodically (every month to six months) look up the root servers to see if the information needs to be altered or augmented.

Local Basic or Advanced Web UI

On the Edit Local CDNS Server tab, under the Root Name Servers category, enter the domain name and IP address of each additional root nameserver, clicking **Add Root Namerserver** after each one, then click **Save**.

CLI Commands

Use **cdns addRootHint**.

Dynamic Allocation of UDP Ports

The Caching DNS server uses a large number of UDP port numbers, by default approximately 60000 port numbers. These numbers are divided among the processing threads. The large number of port numbers reduce the risk of cache poisoning via Birthday Attacks. The Caching DNS server uses the default pool of UDP ports (2048) and the maximum allowable size of the default pool of UDP ports is 4096.

Currently, Cisco Prime Network Registrar uses the port range from 1024 to 65535. Based on the number of outstanding resolution queries, the Caching DNS server adjusts the pool size by adding or removing ports. The Caching DNS server allocates and releases the UDP ports dynamically when the server is running. If you reload the server, all the UDP ports are released and randomly picked again.

Cisco Prime Network Registrar uses *outgoing-range-avoid* attribute that allows you to define ports or ranges of ports that will be excluded from use by the DNS server when sending queries.

Note You need to ensure that UDP ports needed by other applications are in the port exclusion list. Otherwise, these applications may not be able bind to their port(s) if the DNS server is using the port.

Local Basic or Advanced Web UI

On the Edit Local CDNS Server tab, expand Additional Attributes to view various attributes and their values. For the query-source-port-exclusion-list attribute value, enter a range of ports that need to be excluded. Then click Modify Server.

Setting Advanced Caching DNS Server Properties

You can set these advanced server properties:

- **Maximum memory cache sizes**—See the "Setting Maximum Memory Cache Sizes" section on page 19-9.
- **Network Settings—**See the "Specifying Network Settings" section on page 19-10.
- **Flush cache**—See the "Flushing CDNS Cache" section on page 19-10.
- **Prevent DNS cache poisoning**—See the "Detecting and Preventing DNS Cache Poisoning" section on page 19-11.
- **Handle unresponsive nameservers**—See the "Detecting and Preventing DNS Cache Poisoning" section on page 19-11.

Setting Maximum Memory Cache Sizes

The maximum memory cache size property specifies how much memory space you want to reserve for the DNS in-memory cache. The larger the memory cache, the less frequently the Caching DNS server will need to re-resolve unexpired records.

Local Advanced Web UI

On the Edit Local CDNS Server tab, in the Caching category, set it to the desired value for the RRSet Cache Size (*rrset-cache-size*), then click Save. The default size is 100MB.

To set the size of the message cache, use the Message Cache Size (*msg-cache-size*) attribute. The default value for Message Cache Size is 200 MB. The message cache stores query responses. It should generally be twice the size of the RRSet Cache Size (*rrset-cache-size*).

CLI Commands

- Use **cdns set rrset-cache-size** to set RRSet Cache Size.
- Use **cdns set msg-cache-size** to set Message Cache Size.

Specifying Network Settings

The *listen-ip-version* attribute lets you to choose the ip packets to accept and issue. You can check IPv4, IPv6, both, or none. The *listen-protocol* attribute lets you to choose the packet protocol to answer and issue, UDP, TCP, both, or none.

Flushing CDNS Cache

The Cisco Prime Network Registrar cache flushing function lets you remove all or a portion of cached data in the memory cache of the server.

Local Basic or Advanced Web UI

Step 1 From the **Deploy** menu, choose **CDNS Server** under the **DNS** submenu, to open the Manage DNS Caching Server page.

Step 2 On the Manage DNS Caching Server page, click the Commands link to open the CDNS Command dialog box. There will be two types of cache flushing commands.

- Flush the CDNS cache—allows you to either flush all cache entries for a particular zone or the entire cache if no zone is provided. To remove all data for a specific zone, enter the zone name in the Zone field. To clear the whole cache, leave the Zone field empty.
- The Flush Resource Record—allows you to flush an RR name or an RRSet when the type field is specified.
  - Remove common RR types (A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, NAPTR, and TXT) from a specific domain—enter the required RR name as the FQDN for the Flush Resource Record command and leave the RR type field empty.
  - Remove a specified RR type for a domain—specify the domain in the FQDN field, and the RR type in the RR type field.

Note When no type is specified, the server flushes types A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, TXT, and NAPTR.

CLI Commands

To:

- Remove all cached entries at or below a given domain, use **cdns flushCache** *domain*. If no domain is given, it flushes all RRs in the cache.
- Flush RRs from the cache associated with the given RR name, use **cdns flushName** *name type*. When type is provided, it flushes all entries with the given name and type. If no type if provided, it flushes types A, AAAA, NS, SOA, CNAME, DNAME, MX, PTR, SRV, TXT, and NAPTR.

Detecting and Preventing DNS Cache Poisoning

Cisco Prime Network Registrar enhances the CDNS server performance to address the CDNS related issues such as DNS cache poisoning attacks (CSCsq01298), as addressed in a Cisco Product Security Incident Response Team (PSIRT) document number PSIRT-107064 with Advisory ID cisco-sa-20080708-dns, available at:

http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20080708-dns

DNS Cache Poisoning Attacks

A cache poisoning attack can change an existing entry in the DNS cache as well as insert a new invalid record into the DNS cache. This attack causes a hostname to point to the wrong IP address. For example, let us say that www.example.com is mapped to the IP address 192.168.0.1, and this mapping is present in the cache of a DNS server. An attacker can poison the DNS cache and map www.example.com to 10.0.0.1. If this happens, if you try to visit www.example.com, you will end up contacting the wrong web server.

A DNS server that uses a single static port for receiving responses to forwarded queries are susceptible to malicious clients sending forged responses.

The DNS transaction ID and source port number used to validate DNS responses are not sufficiently randomized and can easily be predicted, which allows an attacker to create forged responses to DNS queries. The DNS server will consider such responses as valid.

Handling DNS Cache Poisoning Attacks

To reduce the susceptibility to the DNS cache poisoning attack, the DNS server randomizes the UDP source ports used for forwarded queries. Also, a resolver implementation must match responses to the following attributes of the query:

- Remote address.
- Local address.
- Query port.
- Query ID.
- Question name (not case-sensitive).
- Question class and type, before applying DNS trustworthiness rules (see [RFC2181], section 5.4.1).

Note The response source IP address must match the query's destination IP address and the response destination IP address must match the query's source IP address. A mismatch must be considered as format error, and the response is invalid.

Resolver implementations must:

- Use an unpredictable source port for outgoing queries from a range (either 53, or > 1024) of available ports that is as large as possible and practicable.
- Use multiple different source ports simultaneously in case of multiple outstanding queries.
- Use an unpredictable query ID for outgoing queries, utilizing the full range available (0 to 65535). By default, CDNS uses about 60000 port numbers.

The **Expert** mode Caching DNS server setting *randomize-query-case*, when enabled, specifies that when sending a recursive query, the query name is pseudo-randomly camel-cased and the response is checked to see if this camel-casing is unchanged. If *randomize-query-case* is enabled and the casing has changed, then the response is discarded. The *randomize-query-case* is disabled by default, disabling this feature.

Local Basic or Advanced Web UI

The DNS server statistics appears on the Statistics tab of the Manage DNS Caching Server Statistics page. The Statistics displays the answers-unwanted values. You can refresh the DNS Caching Server Statistics.

Handling Unresponsive Nameservers

When trying to resolve query requests, Caching DNS servers may encounter unresponsive nameservers. A nameserver may be unresponsive to queries, respond late. This affects the performance of the local DNS server and remote nameservers.

Using Cisco Prime Network Registrar, you can resolve these problems by barring unresponsive nameservers. You can configure a global ACL of unresponsive nameservers that are to be barred, using the *acl-do-not-query* attribute.

When Cisco Prime Network Registrar receives a list of remote nameservers to transmit a DNS query request to, it checks for the name-servers listed in the *acl-do-not-query* list and removes them from this list. Conversely, all incoming DNS requests from clients or other nameservers are also filtered against the *acl-blacklist*. The *acl-blacklist* attribute

Note Using the *acl-do-not-query* does not affect the configuration of communication with certain servers such as forwarders.

Use the *acl-query* attribute to specify which clients are allowed to query the server. By default any client is allowed to query the server. A client that is not in this list will receive a reply with status REFUSED. Clients on the *acl-blacklist* do not get any response whatsoever.

Local Advanced Web UI

On the Edit Local CDNS Caching Server tab, expand **Query Access Control** to view the various attributes and their values. For the Do Not Query (*acl-do-not-query*) attribute value, enter, for example, 10.77.240.73. Then click **Save**.

Caching DNS Domain Redirect

DNS domain redirect enables Internet Service Providers (ISP), enterprises, or organizations to redirect the resolution of DNS name away from known bad domains or non-existing domains (NXDOMAIN) for a specified ACL and/or a domain list. Cisco Prime Network Registrar supports DNS domain and NXDOMAIN redirect to override the caching DNS server response to A or AAAA resource record queries.

To ensure that the caching DNS server redirects queries for non-existing or known bad domains, you can create DNS redirect rules. A domain redirect rule comprises of a priority, an ACL, an action, and a list of domains. The domain redirect rules take precedence over exceptions and forwarders.

Every query to a caching DNS server is first verified against the list of redirect rules in the order of priority. When a resource record query matches the criteria of rule, the specified action is taken. If the resource record query action results for *redirect*, *deny*, *refuse*, the corresponding action is taken. If it is *nxdomain*, the query is performed in the normal process and if it results in an NXDOMAIN status, then it is redirected to the specified destination.

Note The Deny and Refuse rules are applicable to all the queries for the specified domains, while the redirect rules and NXDOMAIN are applicable only to the queries of A and AAAA records.

Local Basic or Advanced Web UI

To add, edit, or view the Domain Redirect Rule:

Step 1 From the **Design** menu, choose **Domain Redirect** under the **Cache DNS** submenu to open the List/Add CDNS Domain Redirect Rules page.

Step 2 Click the **Add CDNS Domain Redirect Rule** icon in the Domain Redirect pane to open the Add CDNS Domain Redirect dialog box.

Step 3 Enter a rule name in the **Rule Name** field, and an ACL list in the **Acl List** field.

Step 4 Choose an action from the below list:

- deny—To ignore the resource record query.
- refuse—To block a resource record query.
- redirect—To override the caching DNS response for known bad domains and redirect it to a specified IP address.
- nxdomain—To override the caching DNS response for domains that were not found and redirect to a specified IP address.

Note The rules with the actions Deny and Refuse do not use a destination IP.

Step 5 Click **Add CDNS Domain Redirect** to save the redirect rule. The **List/Add CDNS Domain Redirect Rules** page appears with the newly added redirect rule.

Step 6 Enter the domains that have to be monitored for the redirection.

Note The NXDOMAIN action do not takes domain list.

Step 7 Enter the IPv4 and IPv6 destinations IP addresses.

Step 8 Click **Save** to save your settings, or **Revert** to cancel the changes.

To delete an domain redirect rule, select the rule on the Domain Redirect pane, click the **Delete** icon, and then confirm or cancel the deletion.

CLI Commands

Use the following cli commands to:

- Add the domain redirect rule, separated by spaces, use **cdns-redirect** *rule-name* **create**.

- List the domains the domain redirect rule, use **cdns-redirect list [-priority]**. The priority is optional. It lists the redirect entries according to priority instead of the default, alphabetically by name.
- Remove domain redirect rule, use **cdns-redirect** *rule-name* **delete**.

Reorder DNS Domain Redirect Rules

When you create a set of domain redirect rules, you can specify the priority in which order the rules will apply. To set the priority or reorder the rules:

Step 1 From the **Design** menu, choose **Domain Redirect** under the **Cache DNS** submenu to open the **List/Add CDNS Domain Redirect Rules** page.

Step 2 Click the **Reorder Rules** icon in the Domain Redirect pane to open the Reorder dialog box.

Step 3 Set the priority for the DNS domain redirect rules by either of the following methods:

- Select the rule and click the **Move up** or **Move down** icon to reorder the rules.
- Select the rule and click the **Move to** button, and enter the row number to move the rule.

Step 4 Click **Save** to save the reordered list.