Managing DNS Server Properties

This chapter explains how to set the DNS server parameters. Before you proceed with the tasks in this chapter, read Chapter 15, “Managing Zones,” which explains how to set up the basic properties of a primary and secondary zone.

See Also

Managing DNS Servers
Setting DNS Server Properties, page 17-2
Setting Advanced DNS Server Properties, page 17-12
Tuning DNS Properties, page 17-23
Troubleshooting DNS Servers, page 17-24

Managing DNS Servers

You can view its health, statistics, and logs; start, stop, and reload it; run certain commands (see the “Running DNS Commands” section); 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, click DNS, then DNS Server to open the Manage DNS Server page.

See Also

Running DNS Commands, page 17-1
Configuring DNS Server Network Interfaces, page 17-2

Running DNS Commands

Access the commands by using the Run icon ( ) in the Commands column. Clicking the Run icon opens the DNS Server Commands page in the local web UI. Each command has its own Run icon (click it, then click Return when finished):

- **Flush cache**—This command lets you stop the disk cache file from growing. See the “Flushing DNS Cache” section on page 17-16.
- **Force all zone transfers**—A secondary server periodically contacts its master server for changes. See the “Enabling Zone Transfers” section on page 15-16.
- **Scavenge all zones**—Cisco Network Registrar provides a feature to periodically purge stale records. See the “Scavenging Dynamic Records” section on page 28-16.
Setting DNS Server Properties

- **Remove non-authoritative RR set**—Removes nonauthoritative RRs from both in-memory and persistent (nonauthoritative) cache. See the “Removing Cached Records” section on page 16-5.

**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 DNS Server Network Interfaces

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

**Local Advanced Web UI**

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From the Servers menu, choose Manage Servers.</td>
</tr>
<tr>
<td>2</td>
<td>Click the Interfaces icon ( ) for the DNS server to open the Manage DNS 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.</td>
</tr>
<tr>
<td>3</td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>Clicking the name of the configured interface opens the Edit DNS Server Network Interface page, where you can change the address and port of the interface.</td>
</tr>
<tr>
<td>5</td>
<td>Click <strong>Modify Interface</strong> when you are done editing, then click <strong>Return</strong> to return to the Manage Servers page.</td>
</tr>
</tbody>
</table>

**Note**
The IPv6 functionality in DNS requires IPv4 interfaces to be configured except if the DNS server is isolated and standalone (it is its own root and is authoritative for all queries).

Setting DNS Server Properties

You can set properties for the DNS server, along with those you already set for its zones. These include:

- **General server properties**—See the “Setting General DNS Server Properties” section on page 17-3.
- **Forwarders**—See the “Defining Forwarders for DNS Servers” section on page 17-3.
- **Subzone forwarding**—See the “Setting Subzone Forwarding” section on page 17-5.
- **Resolution exception**—See the “Using Resolution Exception” section on page 17-5.
- **Caching-only servers**—See the “Configuring Caching-Only DNS Servers” section on page 17-7.
- **Delegation-only zones**—See the “Specifying Delegation-Only Zones” section on page 17-8.
- **Root name servers**—See the “Defining Root Nameservers” section on page 17-8.
- **Recursive queries**—See the “Enabling Recursive Queries” section on page 17-8.
Setting DNS Server Properties

- **Round-robin server processing**—See the “Enabling Round-Robin” section on page 17-9.
- **Subnet sorting**—See the “Enabling Subnet Sorting” section on page 17-9.
- **Enabling incremental zone transfers**—See the “Enabling Incremental Zone Transfers (IXFR)” section on page 17-10.
- **Changesets and checkpointing**—See the “Changesets and Checkpointing” section on page 17-10.
- **Restricting zone queries**—See the “Restricting Zone Queries” section on page 17-11.
- **Enabling NOTIFY packets**—See the “Enabling NOTIFY” section on page 17-11.

Setting General DNS Server Properties

You can display DNS general server properties, such as the name of the server cluster or host machine and the version number of the Cisco Network Registrar DNS server software. You can change the internal name of the DNS server by deleting the current name and entering a new one. This name is used for notation and does not reflect the official name of the server. Cisco Network Registrar uses the server IP address for official name lookups and for DNS updates (see Chapter 28, “Configuring DNS Update”).

The following subsections describe some of the more common property settings. They are listed in the “Setting DNS Server Properties” section.

### Local Basic or Advanced Web UI

**Step 1**

To access the server properties, choose **DNS Server** from the **DNS** menu. to open the Manage DNS Server page.

**Step 2**

Click the name of the server to open the Edit DNS Server page. (Or, click **Servers**, then **Manage Servers**, then the **Local DNS Server** link to get to the same page.) The page displays all the DNS server attributes.

**Step 3**

Then click **Modify Server** to save the DNS server attribute modifications.

### CLI Commands

Use `dns [show]` to display the DNS server properties.

### Defining Forwarders for DNS Servers

Sites that must limit their network traffic for security reasons can designate one or more servers to be forwarders that handle all off-site requests before the local server goes out to the Internet. Over time, the forwarders build up a rich data cache that can satisfy most requests. Forwarders are useful in that they:

- **Reduce the load on the Internet connection**—Forwarders build up a cache and thus reduce the number of requests sent to external nameservers and improve DNS performance.
- **Improve the DNS response to repeated queries**—The forwarder cache can answer most queries.
- **Handle firewalls**—Hosts that cannot access the root nameservers can send requests to the forwarder that has access.
Restrict the nameserver even more by stopping it from attempting to contact an off-site server. This kind of server uses forwarders exclusively. It answers queries from its authoritative and cached data, but it relies completely on the forwarders for data not in its cache. If the forwarder does not provide an answer, then the slave mode server does not try to contact other servers.

You can set forwarders by using the `forwarders` DNS server attribute. The forwarders are configured in sequence. Cisco Network Registrar contacts each of the forwarder in that order, and if the forwarder fails to respond within the preset value of eight seconds, it moves on to the next until it receives a response, or it gets to the end of the list. It continues this until one of them responds or until it gets to the end of the list.

If there is no response, the next step depends on whether you have `slave-mode` enabled or disabled:

- If it is enabled, the DNS server stops searching and responds that it cannot find the answer.
- If it is disabled, the server sends the query to the designated name servers. In this case, the server performs as if there were no forwarders listed.

You can adjust the eight-second preset value for the forwarding retry interval using the DNS `forward-retry-time` attribute.

These queries are recursive and can require more time for the forwarder to resolve. Hence set this interval to the `request-expiration-time` (preset to 90 seconds) divided by one less than the total number of configured forwarders or resolution exception servers (if there are more than one forwarder or server). See also the “Enabling Recursive Queries” section on page 17-8.

In some cases, you might see that the resolution queue reaches its configured limit. In this case, non-authoritative queries may time out. This could result from a mail server performing anti-spam lookups or a malicious client flooding the network with queries.

By monitoring network traffic or looking at debug data, you can determine if the resolution queue limit is set too low. You can adjust it by setting the DNS server attribute `resolution-queue-max-size` to a higher value (the preset value in this release is 5000 concurrent requests).

You can also set the DNS server `request-expiration-time` to a lower value than its preset value of 90 seconds. For example, to 30 seconds.

Subzone forwarding (see the “Setting Subzone Forwarding” section on page 17-5) and resolution exception (see the “Using Resolution Exception” section on page 17-5) override any forwarders you set.

**Local Basic or Advanced Web UI**

**Step 1** On the Edit DNS Server page, under the Forwarders category, enter the IP addresses of the forwarding servers.

**Step 2** Click Add Forwarder.

**Step 3** Specify whether you want `no-recursion` or `slave-mode` enabled or disabled.

**Note** Enable `no-recursion` attribute to make a DNS server serve authoritative data alone.

**Step 4** Click Modify Server.
Step 5

Adjust the forward-retry-time and request-expiration-time values if necessary under the Miscellaneous Options and Settings category.

---

**CLI Commands**

To:

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

---

**Setting Subzone Forwarding**

For zones with forwarders set (see the “Defining Forwarders for DNS Servers” section on page 17-3), the normal behavior is to ignore delegation to subzone nameservers and forward queries to these forwarding servers.

This is the case when the zone attribute `subzone-forward` is set to `normal`, which is the preset value in Edit Zone page. To enable delegation to the subzone nameservers under these conditions, you could set a resolution exception (see the “Using Resolution Exception” section) to the subzone server. But, this might be impractical for large numbers of subzones.

Instead, you can set `subzone-forward` to `no-forward`. In this case, when the server receives a query for a subzone, it tries to find the subzone nameserver (NS) record, resolve its IP addresses, and delegates the query to the subzone nameserver. Subzone forwarding overrides any requested forwarders.

---

**Using Resolution Exception**

If you do not want the DNS server to use the standard resolution method to query the root nameserver for certain names outside its domain, use resolution exception. 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 DNS server knows that it is not authoritative for Blue and appeals to 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.

Resolution exception solves 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 of a root server.

To enable resolution exception, set the DNS attribute `exceptions` to the value of one or more nameservers. In this case, recursion applies, the DNS server forwards only once for each query and always in the list order, and the server tries to reach each exception server within the DNS server forward-retry-time. Resolution exception overrides any requested forwarders.
Normally, because the DNS exception-forwarding attribute is preset to **forward-last**, the DNS server forwards queries to the specified resolution exception server or servers only if there is no NS record for the requested domain in its cache.

If there is an NS record, it would always try to forward to that nameserver. You can override this by setting **exception-forwarding** to either of the following:

- **forward-always**—Always forwards queries to the specified exception server or servers.
- **forward-first**—Tries to forward queries to any exception server first, then to any NS nameserver.

Because exception forwarding is set at the server level, it applies to all zones for the server.

In other words, you can have only one behavior for all resolution exceptions. In addition, only those queries that fall into the configured resolution exceptions are forwarded.

With no resolution exception defined, when global forwarding is set (see the “Setting Subzone Forwarding” section on page 17-5), any query for the subzone delegation goes to the forwarder, and not to the server that is authoritative for that subzone.

However, if you set the zone **subzone-forward** attribute to **no-forward**, the zone server is set as the implicit resolution exception server for all the subzones, and any forwarding and explicitly set resolution exceptions are ignored for that zone.

Although defining a resolution exception at a subzone name is a useful way to prevent queries to a specific subzone from being forwarded to the global forwarder. To do this for all subzones of a zone, use subzone forwarding.

**Local Basic or Advanced Web UI**

On the Edit DNS Server page, under the Resolution Exceptions category:

- In the **exceptions** fields, enter the domain name and IP address (or addresses) of each nameserver in the Name and IP Address(es) fields. Click **Add Exception** for each one.
- In the **exception-forwarding** field, accept the **forward-last** preset value, or choose **forward-always** or **forward-first** from the drop-down list. Click **Modify Server**.

To delete an exception list, click the Delete icon (🗑️) next to the domain name and IP address or addresses of the nameserver you want to remove, then click **Modify Server**.

**CLI Commands**

To:

- Add the exception domains and servers, separated by spaces, use **dns listExceptions** and **dns addException**. Use this command only if you do not want your DNS server to use the standard name resolution for names outside the local authoritative zone.
- Guarantee that the server forwards queries to the resolution exception server list instead of (or before) forwarding to the nameserver specified by the NS RR, use **dns set exception-forwarding=forward-always** or **dns set exception-forwarding=forward-first**. Otherwise, use **dns set exception-forwarding=forward-last** (the preset value).
- Remove a resolution exception, use **dns removeException**.
- Replace a resolution exception, use **dns addException** with the new values. You must also flush the cache so that the server does not refer to the old resolution values in cache.
Configuring Caching-Only DNS Servers

By definition, all servers are caching servers, because they save the data that they receive until it expires. However, you can create a caching-only server that is not authoritative for any zone. The only function of this type of server is to answer queries by storing in its memory data from authoritative servers. The caching-only server can then learn or cache the data to answer subsequent queries. Caching query responses considerably reduces the overhead (and latency) of subsequent queries to these cached records.

When you first install Cisco Network Registrar, the DNS server automatically becomes a non-authoritative, caching-only server until you configure zones for it. If you keep the DNS server as a caching-only server, you must have another primary or secondary DNS server somewhere that is authoritative and to which the caching-only server can refer. The caching-only server should never be listed as an authoritative server for a zone.

You must set up a caching-only server to respond to recursive queries, where a server keeps trying to get to an authoritative server so that it can update its cache with the address resolution data. Cisco Network Registrar servers are recursive by default.

To tune the performance of a caching-only server:

- Increase the in-memory cache to the maximum possible value that is aligned to the operating system physical memory capacity by setting the `mem-cache-size` value. The preset value is 10000, but it is recommended to increase this value, for example, to 30000 or 100000.

- When using a large cache, you might want to disable persisting the cache, so that reload time is not impacting the need to save the large cache. Disable the `persist-mem-cache` attribute. (However, note that disabling the attribute discards the cache at each server reload.)

  In cases where you do not want to disable `persist-mem-cache` (such as when the DNS server requires frequent reloads or restarts), there can still be a performance benefit from not writing cached entries with small TTLs. These entries are likely to expire before the next time the server needs them. Therefore, it is recommended not to write cached entries with TTLs of less than 5 seconds. To do this, enter Expert mode in the web UI or use `session set visibility=3` in the CLI, then set the `cache-write-ttl-threshold` attribute to 5s.

- Restrict queries to certain clients only by setting the `restrict-query-acl` attribute.

Reload the DNS server after changing any of these values.

Local Basic or Advanced Web UI

On the Edit DNS Server page, click Show A-Z View, then find each previously described attribute to set it. Reload the server after the changes.

CLI Commands

Use commands similar to the following:

```
nrcmd> dns set mem-cache-size=30000
nrcmd> dns disable persist-mem-cache
nrcmd> session set visibility=3
nrcmd> dns set cache-write-ttl-threshold=5s
nrcmd> session set visibility=5
nrcmd> dns set restrict-query-acl=myaccesslist
nrcmd> dns reload
```
Specifying Delegation-Only Zones

You can instruct the server to expect only referrals when querying the specified zone. In other words, you want the zone to contain only NS records, such as for subzone delegation, along with the apex SOA record of the zone. This can filter out “wildcard” or “synthesized” data from authoritative nameservers whose undelegated (in-zone) data is of no interest. Enable the DNS server `delegation-only-domains` attribute for this purpose.

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 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 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. Because Cisco Network Registrar has a persistent cache, it does not need to requery this data when it restarts. The time to live (TTL) on the official root server records is currently six days, so Cisco Network Registrar requeries every six days, unless you specify a lower maximum cache TTL value (see the “Setting Maximum Cache TTLs” section on page 17-15).

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 DNS Server page, under the Root Nameservers category, enter the domain name and IP address of each additional root nameserver, clicking Add Root Nameserver after each one, then click Modify Server.

CLI Commands

Use `dns addRootHint`.

Enabling Recursive Queries

There are two types of queries—recursive and iterative (nonrecursive). DNS clients typically generate recursive queries, where the nameserver asks other DNS servers for any non-authoritative data not in its own cache. With an iterative query, the nameserver answers the query if it is authoritative for the zone, has the answer in its cache, or tells the client which nameserver to ask next. You often want to make a root server iterative instead of recursive. Recursion is like saying, “Let me talk to Bob and get back to you.” Iteration is like saying, “Let me direct you to Bob for the information.”

You can also restrict the clients from which to honor recursive queries by setting the `restrict-recursion-acl` attribute to an access control list (ACL); see the “Restricting Zone Queries” section on page 17-11.

Here is a list of other attributes that you can set to optimize recursive query performance, although you would generally leave these values to their defaults:

- `forward-retry-time`—Retry interval for forwarding queries to forwarders or resolution exception servers (preset value 8 seconds); see the “Defining Forwarders for DNS Servers” section on page 17-3.
Setting DNS Server Properties

- **request-expiration-time**—Expiration time of general DNS queries (preset value 90 seconds).
- **request-retry-time**—Retry time interval for general DNS queries (preset value 4 seconds).
- **tcp-query-retry-time**—Retry time for DNS queries over TCP, in response to truncated UDP packets (preset value 10 seconds).

**Enabling Round-Robin**

A query might return multiple A records for a nameserver. To compensate for most DNS clients starting with, and limiting their use to, the first record in the list, you can enable *round-robin* to share the load. This method ensures that successive clients resolving the same name will connect to different addresses on a revolving basis. The DNS server then rearranges the order of the records each time it is queried. It is a method of load sharing, rather than load balancing, which is based on the actual load on the server.

**Tip**

Adjust the switchover rate from one round-robin server to another using the TTL property of the server A record.

**Local Basic or Advanced Web UI**

On the Edit DNS Server page, in A-Z view, find the *round-robin* attribute; set it to enabled, then click Modify Server.

**CLI Commands**

Use `dns get round-robin` to see if round-robin is enabled (it is by default). If not, use `dns enable round-robin`.

**Enabling Subnet Sorting**

If you enable subnet sorting, as implemented in BIND 4.9.7, the Cisco Network Registrar DNS server confirms the client network address before responding to a query. If the client, server, and target of the query are on the same subnet, and the target has multiple A records, the server tries to reorder the A records in the response by putting the closest address of the target first in the response packet. DNS servers always return all the addresses of a target, but most clients use the first address and ignore the others.

If the client, DNS server, and target of the query are on the same subnet, Cisco Network Registrar first applies round-robin sorting and then applies subnet sorting. The result is that if you have a local response, it remains at the top of the list, and if you have multiple local A records, the server cycles through them.

**Local Basic or Advanced Web UI**

On the Edit DNS Server page, in A-Z view, find the *subnet-sorting* attribute, set it to enabled, then click Modify Server.

**CLI Commands**

Use `dns enable subnet-sorting` or `dns disable subnet-sorting` (the preset value).
Enabling Incremental Zone Transfers (IXFR)

Incremental Zone Transfer (IXFR, described in RFC 1995) allows only changed data to transfer between servers, which is especially useful in dynamic environments. IXFR works together with NOTIFY (see the “Enabling NOTIFY” section on page 17-11) to ensure more efficient zone updates.

Primary zone servers always provide IXFR. You should explicitly enable IXFR on the server (you cannot set it for the primary zone) only if the server has secondary zones. The DNS server setting applies to the secondary zone if there is no specific secondary zone setting.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the `ixfr-enable` attribute, then set it to enabled. For a secondary zone, you can also fine-tune the incremental zone transfers by setting the `ixfr-expire-interval` attribute.

This value is the longest interval the server uses to maintain a secondary zone solely from IXFRs before forcing a full zone transfer (AXFR). The preset value of one week is usually appropriate. Then, click Modify Server.

CLI Commands

Use `dns enable ixfr-enable`. By default, the `ixfr-enable` attribute is enabled.

Changesets and Checkpointing

Cisco Network Registrar maintains a changeset database that collects recent changes to RR data before they are committed to the authoritative database (auth.db). The DNS server checks the changeset database first before it checks the auth.db when responding to or performing incremental zone transfers or zone checkpointing (see later in this section). A single DNS changeset can consist of one or more RRs.

The changeset database is backed up during the usual `cnr_shadow_backup` backup. To keep it from growing without bounds, the server trims it periodically. The effect of this is, for example, that if a DNS client requests a zone IXFR based on a serial number for RRs that were trimmed, the DNS server cannot perform an IXFR, but must perform a full zone transfer (AXFR).

Zone checkpointing creates a snapshot of the zone data that may be necessary to recreate the auth.db in case it is unstable. The server automatically updates the checkpoint files periodically. In addition to occurring with each changeset that is over a certain size threshold, zone checkpointing happens by default every three hours. You can adjust this interval, from between one and 168 hours, using the `checkpoint-interval` DNS server or zone attribute.

Local Basic or Advanced Web UI

To manually force a zone checkpoint, click the Run icon (_enqueue_operation) on the List/Add Zones or List/Add Reverse Zones page. On the Zone Commands for Zone page, click the Run icon (enqueue_operation) next to Checkpoint zone.

CLI Commands

Use `zone name chkpt`, or dump the zone checkpoint file in a more humanly readable form by using `zone name dumpchkpt`. 
Restricting Zone Queries

You can restrict clients to query only certain zones based on an access control list (ACL). An ACL can contain source IP addresses, network addresses, TSIG keys (see the “Transaction Security” section on page 28-9), or other ACLs. The ACL set by the DNS server restrict-query-acl attribute serves as a filter for non-authoritative queries. If a query targets an authoritative zone, the zone ACL applies.

You can also restrict the server to honor recursive requests from certain clients only by setting the restrict-recursion-acl attribute to an ACL that includes these clients (see “Access Control Lists” section on page 28-8). If you unset restrict-recursion-acl, anyone can request a recursive query, by default.

You can also set the corresponding no-recurse DNS attribute to enable or disable recursion altogether. Only when no-recurse is disabled (the preset value) does the restrict-recursion-acl setting apply.

Enabling NOTIFY

The NOTIFY protocol, described in RFC 1996, lets the Cisco Network Registrar DNS primary server inform its secondaries that zone changes occurred. The NOTIFY packet does not indicate the changes themselves, just that they occurred, and this triggers a zone transfer request. Use NOTIFY in environments where the namespace is relatively dynamic.

Because a zone master server cannot know specifically which secondary server transfers from it, Cisco Network Registrar notifies all nameservers listed in the zone NS records. The only exception is the server named in the SOA primary master field.

You can use IXFR and NOTIFY together, but this is not necessary. You can disable NOTIFY for a quickly changing zone for which immediate updates on all secondaries does not warrant the constant NOTIFY traffic. Such a zone might benefit from having a short refresh time and a disabled NOTIFY.

Local Basic or Advanced Web UI

Step 1 On the Edit DNS Server page, in A-Z view, find the notify attribute, then set it to enabled.
Step 2 Set any of the other NOTIFY attributes (notify-defer-cnt, notify-min-interval, notify-recv-interval, notify-send-stagger, notify-source-address, notify-source-port, and notify-wait).
Step 3 Click Modify Server.
Step 4 To add nameservers in addition to those specified in NS records, click Forward Zones.
Step 5 Click the zone name.
Step 6 Add a comma-separated list of servers using the notify-set attribute on the Edit Zone page.
Step 7 Set the notify attribute to true.
Step 8 Click Modify Zone on that page.

CLI Commands

Use dns enable notify. NOTIFY is enabled by default. You can also enable NOTIFY at the zone level, where you can use zone name set notify-set to specify an additional comma-separated list of servers to notify beyond those specified in NS records.
Setting Advanced DNS Server Properties

You can set these advanced server properties:

- **SOA time-to-live**—See the “Setting SOA Time to Live” section on page 17-12.
- **Secondary server attributes**—See the “Setting Secondary Refresh Times” section on page 17-13.
- **Glue records**—See the “Fetching Glue Records” section on page 17-14.
- **Report lame delegation**—See the “Reporting Lame Delegation” section on page 17-14.
- **Cache**—See the “Setting Maximum Negative Cache Times” section on page 17-15.
- **Prevent rogue A record queries**—See the “Handling Rogue Address Records and Other Cache Attributes” section on page 17-17.
- **Query sources**—See the “Setting Query Source Addresses and Port Numbers” section on page 17-17.
- **Port numbers**—See the “Setting Local and External Port Numbers” section on page 17-18.
- **Handle Malicious DNS Clients**—See the “Handling Malicious DNS Clients and Unresponsive Nameservers” section on page 17-18
- **Handling DNS Cache Poisoning**—See the “Detecting and Preventing DNS Cache Poisoning” section on page 17-20
- **UDP Ports**—See the “Dynamic Allocation of UDP Ports” section on page 17-21

Setting SOA Time to Live

The SOA record time to live (TTL) is usually determined by the zone default TTL. However, you can explicitly set the SOA TTL, which sets the maximum number of seconds a server can cache the SOA record data. For example, if the SOA TTL is set for 3600 seconds (one hour), an external server must remove the SOA record from its cache after an hour and then query your nameserver again.

Cisco Network Registrar responds to authoritative queries with an explicit TTL value. If there is no explicit TTL value, it uses the default TTL for the zone, as set by the value of the `defttl` zone attribute. Databases originating from versions of Cisco Network Registrar earlier than Release 3.5 do not have the `defttl` zone attribute, and use the minimum TTL in the zone SOA record for the default TTL.

Normally, Cisco Network Registrar assumes the default TTL when responding with a zone transfer with RRs that do not have explicit TTL values. If the default TTL value for the zone is administratively altered, Cisco Network Registrar automatically forces a full zone transfer to any secondary DNS server requesting a zone transfer.

Local Basic or Advanced and Regional Web UI

**Step 1**
On the Add Zone or Edit Zone page, set the Zone Default TTL, which defaults to 24 hours.

**Step 2**
If you want, set the SOA TTL, which is the TTL for the SOA records only. It defaults to the Zone Default TTL value.

**Step 3**
You can also set a TTL value specifically for the NS records of the zone. Set the NS TTL value under Nameservers. This value also defaults to the Zone Default TTL value.

**Step 4**
Click Modify Zone.
CLI Commands

Use `zone name set defttl`.

**Setting Secondary Refresh Times**

The secondary refresh time is how often a secondary server communicates with its primary about the potential need for a zone transfer. A good range is from an hour to a day, depending on how often you expect to change zone data.

If you use NOTIFY, you can set the refresh time to a larger value without causing long delays between transfers, because NOTIFY forces the secondary servers to notice when the primary data changes. For details about NOTIFY, see the “Enabling NOTIFY” section on page 17-11.

**Local Basic or Advanced and Regional Web UI**

On the Add Zone or Edit Zone page, set the Secondary Refresh field to the refresh time, which defaults to three hours. Make any other changes, then click **Modify Zone**.

**CLI Commands**

Use `zone name set refresh`. The preset value is 10800 seconds (three hours).

**Setting Secondary Retry Times**

The DNS server uses the secondary retry time between successive failures of a zone transfer. If the refresh interval expires and an attempt to poll for a zone transfer fails, the server continues to retry until it succeeds. A good value is between one-third and one-tenth of the refresh time. The preset value is one hour.

**Local Basic or Advanced and Regional Web UI**

On the Add Zone or Edit Zone page, set the Secondary Retry field to the retry time, which defaults to one hour. Make any other changes, then click **Modify Zone**.

**CLI Commands**

Use `zone name set retry`.
Setting Secondary Expiration Times

The secondary expiration time is the longest time a secondary server can claim authority for zone data when responding to queries after it cannot receive zone updates during a zone transfer. Set this to a large number that provides enough time to survive extended primary server failure. The preset value is seven days.

Local Basic or Advanced and Regional Web UI

On the Add Zone or Edit Zone page, set the Secondary Expire field to the expiration time, which defaults to seven days. Make any other changes, then click Modify Zone.

CLI Commands

Use `zone name set expire`.

Fetching Glue Records

A glue record is a DNS A record that specifies the address of the authoritative nameserver of a zone or subzone. It is an informational record in a query response. For example, most answers include NS records, which then cause the inclusion of A records to resolve the NS record name into an address. These A records are the glue records. Disabling the `no-fetch-glue` attribute tells the server to find records that it would not normally find, so that it can include them in answers to subsequent queries.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the `no-fetch-glue` attribute, set it to disabled, then click Modify Server.

CLI Commands

Use `dns disable no-fetch-glue` (the preset value).

Reporting Lame Delegation

Lame delegation occurs when a DNS server listed in the parent delegation of a zone does not know that it is authoritative for the zone. The server can detect and report this when, in the process of tracking down an answer, it is referred to a server that in turn refers to another server for a domain closer to the root (actually farther from the answer). Lame delegation does not indicate a problem with the DNS configuration, but with the configuration at the DNS server you are querying. You cannot do anything to correct lame delegation at other domains, unless you happen to be authoritative for the domain as well.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, under `log-settings`, check the `lame-delegation` value check box, then click Modify Server.

CLI Commands

Use `dns set log-settings=lame-delegation` (one of the preset values).
Setting Maximum Negative Cache Times

To ensure a quick response to repeated requests for the same information, the DNS server maintains a cache of data it learns from other servers on behalf of its clients. This includes negative information, such as “no such name” or “no such data,” as specified by RFC 2308. It is important to discard this information at some point to accommodate namespace changes at the authoritative source.

The maximum negative cache time establishes an upper bound on the time a negative cache entry can be valid. This value should be obtained from the SOA record in the negative response. If the originating zone has an abnormally large TTL value, the maximum negative cache time value reduces that value, limiting the time that the entry remains negatively cached. Choose this value carefully to balance the need to eliminate long negative cache entries with the desire to cache negative answers meaningfully.

Note that you can effectively reduce the maximum negative cache time through the maximum cache TTL value, because this value applies to all cache entries, positive and negative. See the “Setting Maximum Cache TTLs” section on page 17-15. The smaller of the TTL values of the max-cache-ttl and max-negcache-ttl wins when caching negative entries.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the max-negcache-ttl attribute, set it to the desired value (the preset value is one hour), then click Modify Server.

CLI Commands

Use dns set max-negcache-ttl (the preset value is 60 minutes).

Setting Maximum Cache TTLs

The Maximum Cache TTL property specifies the maximum time that you want Cisco Network Registrar to retain cached information. 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. This parameter limits the lifetime of records in the cache whose TTL values are very large. The preset value is seven days (604800 seconds).

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the max-cache-ttl attribute, set it to the desired value (the preset value is one week), then click Modify Server.

CLI Commands

Use dns set max-cache-ttl.
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 more memory that you allocate for the cache, the less frequently the server uses disk cache. One cache entry is about 100 bytes. The preset value prior to Cisco Network Registrar 6.0 was 200 KB; the value as of 6.0 is 10000 KB (10 MB). The recommended value is 30000 (30 MB).

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the mem-cache-size attribute, set it to the desired value, then click Modify Server. The recommended value is 30000 (30 MB).

CLI Commands

Use `dns set mem-cache-size`.

Flushing DNS Cache

The Cisco Network Registrar cache flushing function lets you stop the disk cache file from growing. However, the actual behavior depends on whether the DNS server is running or stopped. If you flush the cache:

- While the server is running, Cisco Network Registrar clears all expendable entries from the cache database file. Flushing the cache does not shrink the file because of the nature of the database, but does create free space in it. Because the memory cache is unaffected by this operation, recently used cache entries are not lost, and performance is not significantly affected.
- With the server stopped, Cisco Network Registrar interprets the request to flush all entries and removes the cache file. It then reinitializes the database when you restart the server.

To clear a cache that grew too large, or when changing a resolution exception, reload the dns server and enter the command. Stopping the server does not terminate the server process, but stops it from handling further requests. For details on resolution exception, see the “Using Resolution Exception” section on page 17-5.

Local Basic or Advanced Web UI

On the Manage DNS Server page, click the Run icon ( ) in the Commands column to open the DNS Server Commands page (see Figure 7-1 on page 7-2). On this page, click the Run icon ( ) next to Flush cache. If you want to flush cache older than a certain time, enter the time in the Time field next to Flush cache. If you want to flush cache older than, then click the Run icon.

CLI Commands

To flush the cache, use `dns flushCache`.

```
nrcmd> dns reload
nrcmd> dns flushCache
```
Handling Rogue Address Records and Other Cache Attributes

You may become victim of a suspicious denial-of-service attack where a rogue host targets Address (A) RR queries to a caching DNS server. These A record queries contain names that resemble IP addresses. To avoid overloading the DNS server cache.db file with negative responses from the root, the server no longer tries to resolve these queries. The fake-ip-name-response DNS attribute (Fake Responses for IP address-like names in the web UI) is enabled by default to effect this. When the server receives a query for a non-authoritative name, it consults its cached data. If the server cannot answer from cached data, it examines the queried name to see if it resembles an IP address (four decimals each separated by a dot with no trailing or preceding characters). If so, it does not forward the query. Instead, the server sends a response with a NAME ERROR (NXDOMAIN) return code, and does not cache this synthesized negative response.

The server acts on the save-negative-cache-entries (Save negative cache entries to disk in the web UI) and cache-write-til-threshold attributes when it evicts entries from its memory cache to the cache.db file. It typically evicts positive and negative query responses from in-memory cache when the in-memory cache is full and the server needs to inserts a new entry. The server evicts the least-recently-used entry. If you disable save-negative-cache-entries, the server does not store evicted negative entries in the cache.db file, but simply discards them from the in-memory server cache. If the cache-write-til-threshold value is nonzero (it is zero by default), the server only persists entries from the in-memory cache to the cache.db file if the TTLs of an entry are greater than this value. Otherwise, the server discards the (soon to be, but not yet, expired) RR. A zero value causes the server to always persist unexpired RRs.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the fake-ip-name-response and save-negative-cache-entries attributes, set them to enabled, then click Modify Server.

CLI Commands

Use dns enable fake-ip-name-response and dns enable save-negative-cache-entries.

Setting Query Source Addresses and Port Numbers

The query source address is the source IP address from which the DNS server should send queries to other servers when resolving names for clients. A value of 0.0.0.0 indicates that the best local address is used based on the destination.

The query source port determines which ports are used by the DNS server to send queries to other servers when resolving names for clients. The default value of zero indicates that the UDP port pool is a collection of random ports. A non-zero value indicates that the UDP port pool will be a pool of sequential ports, which starts with the value given by the query source port. (see the “Setting Local and External Port Numbers” section).

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the query-source-address and query-source-port attributes, set them to the desired values (there are none preset), then click Modify Server.

CLI Commands

Use dns set query-source-address and dns set query-source-port.
Setting Local and External Port Numbers

If you are experimenting with a new group of nameservers, you might want to use nonstandard ports for answering requests and asking for remote data. The local port and external port settings control the TCP and UDP ports on which the server listens for name resolution requests, and to which port it connects when making requests to other nameservers. The standard value for both is port 53. If you change these values during normal operation, the server will appear to be unavailable.

The full list of default ports is included in the “Default Ports for Cisco Network Registrar Services” section on page 1-12.

Local Basic or Advanced Web UI

On the Edit DNS Server page, in A-Z view, find the local-port-num and remote-port-num attributes, set them to the desired values (they are both preset to 53), then click Modify Server.

Handling Malicious DNS Clients and Unresponsive Nameservers

When trying to resolve query requests, DNS servers may encounter malicious DNS clients or unresponsive nameservers. A client may flood the network with suspicious DNS requests, while a nameserver may be unresponsive to queries, respond late, or consistently provide lame-delegation responses. Both problems affect performance of the local DNS server and remote nameservers.

Using Cisco Network Registrar, you can resolve these problems by barring malicious clients and unresponsive nameservers. You can configure a global ACL of malicious clients and unresponsive nameservers that are to be barred, using the blackhole-acl attribute.

When Cisco Network Registrar receives a list of remote nameservers to transmit a DNS query request to, it checks for the blackholed name-servers and removes them from this list. Conversely, all incoming DNS requests from clients or other name-servers are also filtered against this blackhole-acl.

Note

Using the blackhole-acl does not affect the configuration of communication with certain servers such as forwarders.
Local Basic or Advanced Web UI

On the Edit DNS Server page, expand **Miscellaneous Options and Settings** to view various attributes and their values. For the blackhole-acl attribute value, enter, for example, 10.77.240.73. Then click **Modify Server** (see **Figure 17-1**).

**Figure 17-1  Miscellaneous Options and Settings (Local Advanced)**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Data Type</th>
<th>Default</th>
<th>Unset?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get/verify missing glue records</td>
<td></td>
<td>boolean</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>Use network</td>
<td></td>
<td>boolean</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>Enable broadcast</td>
<td></td>
<td>boolean</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>Use reverse</td>
<td></td>
<td>boolean</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>Max. recursive record caching TTL</td>
<td></td>
<td>bounded int</td>
<td>1W</td>
<td></td>
</tr>
<tr>
<td>Max. recursive record TTL</td>
<td></td>
<td>bounded int</td>
<td>60m</td>
<td></td>
</tr>
<tr>
<td>Default recursive TTL</td>
<td></td>
<td>bounded int</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Max. recursive from site</td>
<td></td>
<td>bounded integer</td>
<td>50000</td>
<td></td>
</tr>
<tr>
<td>Persistent infinity</td>
<td></td>
<td>boolean</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Control recursion acl</td>
<td></td>
<td>boolean</td>
<td>normal</td>
<td></td>
</tr>
<tr>
<td>Max Dns-Packets</td>
<td></td>
<td>unspec'd</td>
<td>533</td>
<td></td>
</tr>
<tr>
<td>Forward retry-time</td>
<td></td>
<td>bounded int</td>
<td>6s</td>
<td></td>
</tr>
<tr>
<td>User-forward retry-time</td>
<td></td>
<td>bounded int</td>
<td>10s</td>
<td></td>
</tr>
<tr>
<td>Request retry-time</td>
<td></td>
<td>bounded int</td>
<td>6s</td>
<td></td>
</tr>
<tr>
<td>Request minimum-time</td>
<td></td>
<td>bounded int</td>
<td>30s</td>
<td></td>
</tr>
<tr>
<td>Request minimum-time</td>
<td></td>
<td>bounded int</td>
<td>15s</td>
<td></td>
</tr>
<tr>
<td>Blackhole-acl</td>
<td></td>
<td>address match</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

**CLI Commands**

Using **dns set blackhole-acl-IP_address**, configure a blackhole-acl list by specifying the IP addresses of the servers that you want to bar.

For example:

```
nrcmd> dns set blackhole-acl="10.77.240.67, acl, 10.77.240.73"
```

Using **dns get blackhole-acl**, view the list of servers that are part of the blackhole-acl.

For example:

```
nrcmd> dns get blackhole-acl
```
Detecting and Preventing DNS Cache Poisoning

Cisco Network Registrar enhances the DNS server performance to address the DNS 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:


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 the 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).
The DNS server setting 'no-tweak-0x20', when false, 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 'no-tweak-0x20' is enabled and the casing has changed, then the response is discarded. The 'no-tweak-0x20' is true by default, disabling this feature.

**Note**

Source ports could be selected from a larger range than indicated, but that this range is regarded as safe, with some lower port numbers reserved for activities which might conflict with proper DNS operation.

**Local Basic or Advanced Web UI**


**CLI Commands**

Use `dns show response-mismatches` and `dns show response-mismatch-status`.

**Dynamic Allocation of UDP Ports**

Cisco Network Registrar enhances the capabilities of the DNS server to provide a dynamic pool of UDP ports that are configured during server initialization for resolution queries. The DNS server uses the default pool of UDP ports (2048) and the maximum allowable size of the default pool of UDP ports is 4096 (CSCsu55330).

Currently, Cisco Network Registrar uses the port range from 1024 to 65535. Based on the number of outstanding resolution queries, the DNS server adjusts the pool size by adding or removing ports. The 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 Network Registrar uses `query-source-port-exclusion-list` attribute that allows you to define a list of port ranges that will be excluded from use by the DNS server when it allocates ports for the UDP port pool.

**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 DNS Server page, 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**.

CLI Commands

Use `dns set local-port-num` and `dns set remote-port-num`.

To exclude a range of ports:

```
dns addException query-source-port-exclusion-list <ip addresses>
```

To list the excluded range of ports:

```
dns listExceptions query-source-port-exclusion-list.
```

To remove the ports:

```
dns removeException query-source-port-exclusion-list
```
Tuning DNS Properties

Here are some tips to tune some of the DNS server properties:

- **Notify send min. interval DNS server attribute** (*notify-min-interval in the CLI*)—Minimum interval required before sending notification of consecutive changes on the same zone to a server. The preset value is two seconds. For very large zones, you might want to increase this value to exceed the maximum time to send an outbound full zone transfer. This is recommended for secondary servers that receive inbound incremental zone transfers and send out full transfers to other secondaries. These include older BIND servers that do not support incremental zone transfers. Inbound incremental transfers may abort outbound full transfers.

- **Notify delay between servers DNS server attribute** (*notify-send-stagger in the CLI*)—Interval to stagger notification of multiple servers of a change. The preset value is one second, but you may want to raise it to up to five seconds if you need to support a large number of zone transfers distributed to multiple servers.

- **Notify wait for more changes DNS server attribute** (*notify-wait in the CLI*)—Time to delay, after an initial zone change, before sending change notification to other nameservers. The preset value is five seconds, but you may want to raise it to 15, for the same reason as given for the *notify-min-interval* attribute.

- **Max. memory cache size DNS server attribute** (*mem-cache-size in the CLI*)—Size of the in-memory record cache, in kilobytes. The preset value is 200 KB, but you may want to raise it to 1000 KB for larger networks.

- **Report lame delegations DNS server attribute** (*lame-deleg-notify in the CLI*)—Cisco Network Registrar should notice and log when a DNS server listed in a parent-zone delegation of subzones does not know that it is authoritative for the zone. This is normally disabled, but you may want to enable it. This attribute has the same effect as setting the log settings to *lame-delegation*.

- **Maximum UDP payload size DNS server attribute** (*max-udp-payload-size*)—The maximum UDP payload size of the DNS server that responds to the client. You can modify this attribute from a minimum of 512 bytes to a maximum of 4 KB. The default value for this attribute is set to the maximum, that is, 4 KB on the DNS server.

- **IXFR check box in the Foreign Servers section of the Edit DNS Server page, or remote-dns address/mask create ixfr in the CLI**—Adding an entry for a server or group of servers allows controlling whether or not IXFR should occur when doing zone transfers from those servers.
Troubleshooting DNS Servers

Useful troubleshooting hints and tools to diagnose the DNS server and ways to increase performance include:

- **Restoring a loopback zone**—A loopback zone is a reverse zone that enables a host to resolve the loopback address (127.0.0.1) to the name `localhost`. The loopback address is used by the host to enable it to direct network traffic to itself. You can configure a loopback zone manually or you can import it from an existing BIND zone file.

- **Listing the values of the DNS server attributes**—Click DNS, then DNS Server to open the Edit DNS Server page in the web UI. In the CLI, use `dns show`.

- **Adjusting certain attribute values that could have inherited preset values from previous releases during an upgrade**—For deployments that were upgraded from Cisco Network Registrar 5.5.x or earlier, the DNS server might be operating with legacy preset values for critical settings. These preset values are probably not optimal for current systems and can cause performance issues. We strongly recommend that you update the legacy settings to use the new preset values. Table 17-1 lists the old and new preset values, along with a recommended setting for each attribute.

<table>
<thead>
<tr>
<th>DNS Attribute</th>
<th>7.0 Preset Value</th>
<th>7.1 Preset Value</th>
<th>Recommended Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>auth-db-cache-kbytes</code></td>
<td>5120 (KB)</td>
<td>10240 (KB)</td>
<td>102400 (KB)</td>
</tr>
<tr>
<td><code>axfr-multirec-default</code></td>
<td>on</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td><code>cache-db-cache-kbytes</code></td>
<td>5120</td>
<td>10240</td>
<td>102400 (KB)</td>
</tr>
<tr>
<td><code>changeset-db-cache-size</code></td>
<td>10000 (KB)</td>
<td>10000 (KB)</td>
<td>10000 (KB)</td>
</tr>
<tr>
<td><code>changeset-db-logs-trimming-interval</code></td>
<td>30m</td>
<td>30m</td>
<td>30m</td>
</tr>
<tr>
<td><code>htrim-zone-seek-more-trim-interval</code></td>
<td>5m</td>
<td>5m</td>
<td>5m</td>
</tr>
<tr>
<td><code>mem-cache-size</code></td>
<td>10000 (KB)</td>
<td>50000 (KB)</td>
<td>50000 (KB)</td>
</tr>
<tr>
<td><code>zone-db-cache-kbytes</code></td>
<td>1024 (KB)</td>
<td>1024 (KB)</td>
<td>1024 (KB)</td>
</tr>
<tr>
<td><code>query-source-port</code></td>
<td>53</td>
<td>0</td>
<td>0 (let the operating system choose the outbound DNS port)</td>
</tr>
</tbody>
</table>

For many of these attributes, you must enter Expert mode in the web UI or use `set session visibility=3` in the CLI. To change the preset value to the current one, unset the attribute. To change to the recommended setting, change the attribute value.

Be sure to reload the DNS server after saving the settings.

- **Choosing from the DNS log settings to give you greater control over existing log messages**—Use the Log settings attribute on the Edit DNS Server page in the web UI, or `dns set log-settings` in the CLI, with one or more of these keyword or numeric values, separated by commas (see Table 17-2). Restart the server if you make any changes to the log settings.
### Table 17-2  
**DNS Log Settings**

<table>
<thead>
<tr>
<th>Log Setting (Numeric Equivalent)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config (1)</td>
<td>Server configuration and deinitialization.</td>
</tr>
<tr>
<td>ddns (2)</td>
<td>High level dynamic update messages.</td>
</tr>
<tr>
<td>xfr-in (3)</td>
<td>Inbound full and incremental zone transfers.</td>
</tr>
<tr>
<td>xfr-out (4)</td>
<td>Outbound full and incremental zone transfers.</td>
</tr>
<tr>
<td>notify (5)</td>
<td>NOTIFY transactions.</td>
</tr>
<tr>
<td>datastore (8)</td>
<td>Data store processing that provides insight into various events in the server embedded databases.</td>
</tr>
<tr>
<td>scavenge (9)</td>
<td>Scavenging of dynamic RRs (see the “Scavenging Dynamic Records” section on page 28-16).</td>
</tr>
<tr>
<td>scavenge-details (10)</td>
<td>More detailed scavenging output (disabled by default).</td>
</tr>
<tr>
<td>server-operations (11)</td>
<td>General high-level server events, such as those pertaining to sockets and interfaces.</td>
</tr>
<tr>
<td>lame-delegation (13)</td>
<td>Lame delegation events; although enabled by default, disabling this flag could prevent the log from getting filled with frequent lame delegation encounters.</td>
</tr>
<tr>
<td>root-query (14)</td>
<td>Queries and responses from root servers.</td>
</tr>
<tr>
<td>ddns-refreshes (15)</td>
<td>DNS update refreshes for Windows clients (disabled by default).</td>
</tr>
<tr>
<td>ddns-refreshes-details (16)</td>
<td>RRs refreshed during DNS updates for Windows clients (disabled by default).</td>
</tr>
<tr>
<td>ddns-details (17)</td>
<td>RRs added or deleted due to DNS updates.</td>
</tr>
<tr>
<td>tsig (18)</td>
<td>Logs events associated with Transaction Signature (TSIG) DNS updates (see the “Transaction Security” section on page 28-9).</td>
</tr>
<tr>
<td>tsig-details (19)</td>
<td>More detailed logging of TSIG DNS updates (disabled by default).</td>
</tr>
<tr>
<td>activity-summary (20)</td>
<td>Summary of activities in the server. You can adjust the interval at which these summaries are taken using the activity-summary-interval attribute, which defaults to five-minute intervals (you can adjust this interval using dns set activity-summary-interval).</td>
</tr>
<tr>
<td>query-errors (21)</td>
<td>Logs errors encountered while processing DNS queries.</td>
</tr>
<tr>
<td>config-details (22)</td>
<td>Generates detailed information during server configuration by displaying all configured and assumed server attributes (disabled by default).</td>
</tr>
<tr>
<td>incoming-packets (23)</td>
<td>Incoming data packets.</td>
</tr>
<tr>
<td>outgoing-packets (24)</td>
<td>Outgoing data packets.</td>
</tr>
<tr>
<td>xfer-in-packets (25)</td>
<td>Incoming full zone transfer (XFR) packets.</td>
</tr>
<tr>
<td>query-packets (26)</td>
<td>Incoming query packets.</td>
</tr>
<tr>
<td>notify-packets (27)</td>
<td>NOTIFY packets.</td>
</tr>
<tr>
<td>ddns-packets (28)</td>
<td>DNS Update packets.</td>
</tr>
<tr>
<td>xfer-out-packets (29)</td>
<td>Outgoing XFR packets.</td>
</tr>
<tr>
<td>ha-details (30)</td>
<td>Generates detailed logging of High-Availability (HA) DNS information.</td>
</tr>
</tbody>
</table>
Troubleshooting DNS Servers

- **Using the nslookup utility to test and confirm the DNS configuration**—This utility is a simple resolver that sends queries to Internet nameservers. To obtain help for the nslookup utility, enter `help` at the prompt after you invoke the command. Use only fully qualified names with a trailing dot to ensure that the lookup is the intended one. An nslookup begins with a reverse query for the nameserver itself, which may fail if the server cannot resolve this due to its configuration. Use the `server` command, or specify the server on the command line, to ensure that you query the proper server. Use the `-debug`, or better yet, the `-d2`, flag to dump the responses and (with `-d2`) the queries being sent.

- **Using the dig utility to troubleshoot DNS Server**—dig (domain information groper) is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use, and clarity of output. To obtain help for the dig utility, enter `help` at the prompt after you invoke the command.

  Although dig is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. Unlike earlier versions, the BIND9 implementation of dig allows multiple lookups to be issued from the command line. Unless you specifically query a specific name server, dig tries each of the servers listed in `/etc/resolv.conf`. When no command line arguments or options are given, dig performs an NS query for the root ".". A typical invocation of dig looks like: `dig @server name type` where `server` is the name or IP address of the name server to query.