



# Configuring the CSS Domain Name Service

---

This chapter provides an overview of the CSS Domain Name Service (DNS) feature and describes how to configure it for operation. Information in this chapter applies to all CSS models, except where noted.



## Note

---

The CSS Domain Name Service feature is part of the CSS Enhanced feature set.

---

This chapter provides the following major sections:

- [Overview of the CSS Domain Name Service](#)
- [Overview of the CSS Application Peering Protocol](#)
- [Configuring the Application Peering Protocol](#)
- [Configuring a CSS as an Authoritative DNS Server](#)
- [Displaying DNS Server and Zone Information](#)
- [Configuring Domain Records](#)
- [Displaying DNS Record Information](#)
- [Configuring DNS Using Content Rules](#)
- [Configuring Source Groups to Allow Servers to Resolve Domain Names Using the Internet](#)
- [Displaying Domain Summary Information](#)

# Overview of the CSS Domain Name Service

The CSS Domain Name Service (DNS) feature enables you to configure one or more CSSs together to construct highly available, distributed, and load-sensitive Web sites. Groups of CSSs may host many distributed Web sites concurrently. These groups make decisions that can be configured independently for each distributed Web site using local and remote load-balancing information.

CSSs that are configured together for DNS form a *content domain*. Within the content domain, CSSs are known as peers. You can configure peers to exchange content rules, load-balancing information, and service availability.

Each CSS becomes aware of all the locations for the content associated with a domain name and the operational state and load of the location. The CSS can then intelligently direct clients to a site where they can best obtain the desired content. In addition, a CSS never sends a client to a location that is overburdened or out of service.

You can use DNS to configure a CSS as a DNS authoritative server. A CSS defined as an authoritative DNS server resolves DNS names when requested by a client.

For example, when a user clicks on a URL on a Web page:

1. The client asks the locally configured DNS server for a translation of a domain name to an IP address. The DNS server contains the CSS virtual IP address (VIP) and DNS names.
2. The DNS server requests address resolution from the CSS DNS authoritative server.
3. The CSS DNS authoritative server returns the VIP address of the best location (that is, server availability and load) where the client can retrieve the content.
4. The DNS server responds to the client with the VIP.
5. The client uses the VIP to access the content.

**Note**

---

The CSS implementation of DNS server functionality is a streamlined, endnode-only approach. The CSS does not support zone transfer among other DNS servers. However, each CSS configured in a content domain can act as the authoritative DNS server.

---

# Overview of the CSS Application Peering Protocol

CSSs configured within the same content domain initiate communication over Application Peering Protocol (APP) sessions with their peers upon system bootup or when peers first become connected through an APP session. Thereafter, changes in local configurations are relayed to the peers automatically as they occur. When the APP session is up, the peers exchange owner names according to the DNS exchange policies configured for each owner.

For each owner that a CSS is configured to share with its peers, the CSS sends the locally configured content rules and DNS name information. Upon receiving a peer's content rule information, the CSS compares each DNS name and content rule to its local configuration.

Content rules that:

- Match a locally configured content rule cause a *dynamic service* to be added automatically to the local content rule. The local content rule points to the peer for an alternate location for the content.
- Do not have a corresponding local entry cause the CSS to automatically create a *dynamic content rule* containing a dynamic service that points to the peer that has the content rule configured.

The determination of whether or not a content rule matches is based strictly on content rule name. Peers having matching content rule names must have exact copies of rule definitions with the exception of VIP addresses. DNS names do not need to be identical.



## Note

---

CSSs do not include dynamic services or dynamic content rules in their running- or startup-config files. Dynamic services and dynamic content rules are temporary and are removed when the peer connection terminates.

---

For example, when a client requests *www.arrowpoint.com*:

1. The client browser asks the locally configured DNS server for a translation to an IP address.
2. The DNS server round-robins an address resolution request to one of the CSSs.

3. The selected CSS DNS authoritative server determines server availability based on the DNS balance type.

If the CSS is configured as DNS balance type **dnsbalance preferlocal** and is:

- Able to locally handle the request for this DNS name, it returns the local VIP to the DNS server.
- Not able to handle the request for this DNS name (the server has reached a defined load threshold or is unavailable), the CSS returns the dynamic content rule VIP to the DNS server.

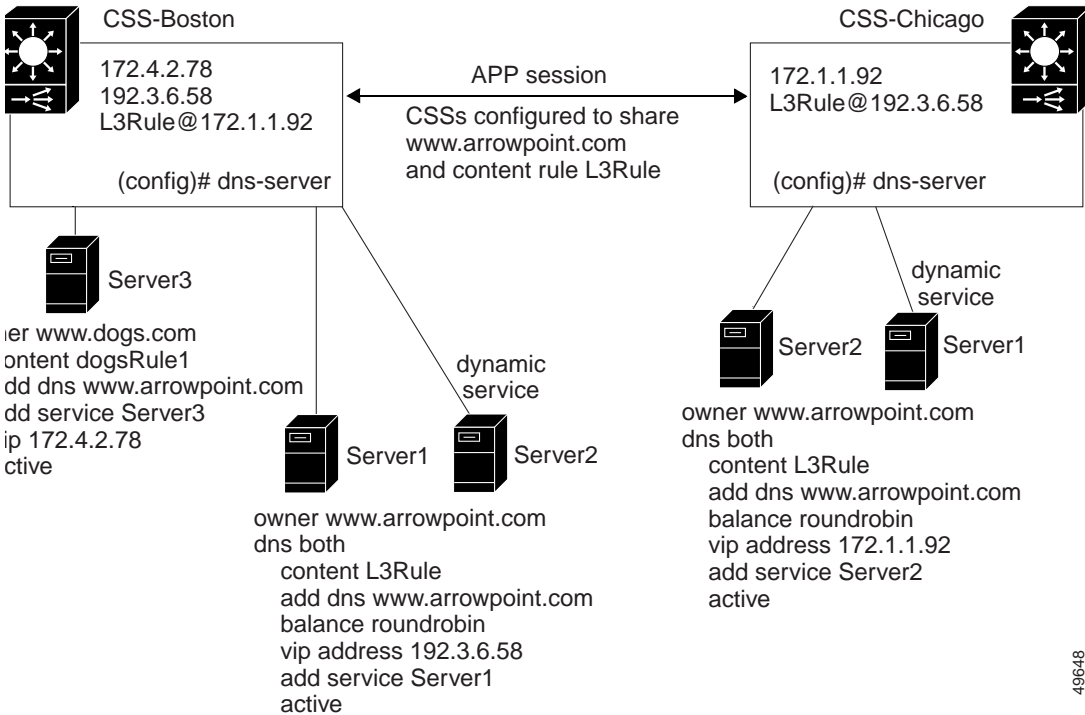
If the CSS is configured as DNS balance type **dnsbalance roundrobin** the CSS resolves requests by evenly distributing the load to resolve domain names among local and remote content domain sites.

For information on configuring DNS balance types, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

4. The DNS server forwards the resolved VIP to the client.

**Figure 1-1** illustrates two peer CSSs configured as authoritative DNS servers. Each CSS knows its local content rule VIPs and dynamic content rule VIPs. The @ sign within a content rule VIP indicates a dynamic content rule. Owner *www.arrowpoint.com* is configured for **dns both** (push and accept owner *www.arrowpoint.com* and its content rule *L3Rule*). Even though CSS-Boston contains owners *www.arrowpoint.com* and *www.dogs.com*, only owner *www.arrowpoint.com* and content rule *L3Rule* are shared between the CSSs.

Figure 1-1 CSS Configured as an Authoritative DNS



## Configuring the Application Peering Protocol

When two CSSs communicate, they use an Application Peering Protocol (APP) session. An APP session allows the exchange of content information between a pair of configured CSSs. APP provides a guaranteed and private communications channel for this exchange. Two or more CSSs that are configured to exchange content rules over APP sessions form a content domain and are considered peers.



### Note

The Application Peering Protocol feature is part of the CSS Enhanced feature set.

To configure APP sessions, use the **app** command. The options for this global configuration mode command are:

- **app** - Enables all APP sessions
- **app framesz** - Sets the maximum frame size allowed by the APP
- **app port** - Sets the TCP port that listens for APP connections
- **app session** - Creates an APP session

For example:

```
(config)# app
```

To disable all APP sessions, enter:

```
(config)# no app
```

## Configuring APP Frame Size

To set the maximum size allowed by the APP, use the **app framesz** command. Enter the maximum APP frame size from 10240 to 65535. The default is 10240. Upon session establishment, peers select the smallest configured frame size to use for session communication. For example, CSS-A is configured for frame size 5000 and CSS-B is configured for frame size 6000. Once the session is established, CSS-B will use frame size 5000.

For example:

```
(config)# app framesz 5096
```

To restore the default frame size to 10240, enter:

```
(config)# no app framesz
```

## Configuring APP Port

To set the TCP port number, use the **app port** command. This port listens for APP connections. Enter a port number from 1025 to 65535. The default TCP port is 5001.

For example:

```
(config)# app port 21
```

To restore the default port number to 5001, enter:

```
(config)# no app port
```

## Configuring an APP Session

To create an APP session between two CSSs, use the **app session** command. The CSSs use APP sessions to create a content domain that shares the same content rules, load, and DNS information with each other.

The syntax and options for this global configuration mode command are:

```
app session ip_address {keepalive frequency {authChallenge|authNone  
session_secret {encryptMd5hash|encryptNone  
{rcmdEnable|rcmdDisable}}}}
```



### Note

The **authChallenge|authNone** and **encryptMd5hash|encryptNone** APP command options must be identical for both CSSs in an APP session or the session will not come up.

The **keepalive** and **rcmd** command options do not have to be identical between CSS peers.

The variables and options are:

- *ip\_address* - IP address for the peer CSS.



### Note

Do not configure an APP session peer with a local CSS IP address (for example, a circuit IP address or Management port IP address). If you do, the following error message appears: `Illegal IP address for APP.`

- *keepalive frequency* - Optional time in seconds between sending keepalive messages to this peer CSS. Enter an integer from 14 to 255. The default is 14.
- **authChallenge|authNone** - Optional authentication method for the session. Enter either **authChallenge** for Challenge Handshake Authentication Protocol (CHAP) method or **authNone** for no authentication method. The default is no authentication.

- *session\_secret* - Secret used with AuthChallenge to authenticate a peer or used with encryptMd5hash to provide an MD5hash encryption scheme for the session. Enter an unquoted text string with a maximum of 32 characters.
- **encryptMd5hash|encryptNone** - Optional encryption method for the packets. Enter either **encryptMd5hash** for MD5 base hashing method or **encryptNone** for no encryption method. The default is no encryption.
- **rcmdEnable|rcmdDisable** - Optional setting for sending remote CLI commands to the peer through the **rcmd** command. Enter either **rcmdEnable** to allow the sending of CLI commands or **rcmdDisable** to disallow the sending of CLI commands. The default setting is enabled.

To terminate an APP session, enter the **no app session** command and an IP address:

```
(config)# no app session 192.2.2.2
```

For example, to configure a CSS in Boston (IP address 172.1.1.1) to be a peer of a CSS in Chicago (IP address 192.2.2.2), use the **app** command to configure:

```
CSS-Boston(config)# app session 192.2.2.2
CSS-Chicago(config)# app session 172.1.1.1
```

## Using the rcmd Command

To issue remote CLI commands to a CSS peer, use the **rcmd** command. Before you can use this command, use the **(config) app session** command to configure an APP session. The **rcmd** command is available in SuperUser mode.

The syntax for this command is:

```
rcmd ip_address or host “CLI command {;CLI command...}”
{timeout_response}
```

The variables are:

- *ip\_address* or *hostname* - The IP address or host name for the peer.
- *CLI command* - One or more CLI commands you want to issue to the peer. Enter the command, its options, and variables exactly. Enclose the command text string in quotes (“”). When entering multiple CLI commands, insert a semicolon (;) character to separate each command.



---

**Note** You cannot issue **grep**, **grep** within a script command, or **redirect** commands.

---

- *timeout\_response* - The optional amount of time, in seconds, to wait for the output command response from the peer. Enter an integer from 3 to 300 (5 minutes). The default is 3 seconds.



---

**Note** By default, the APP session is configured to allow sending remote commands to a CSS peer. If you disable this function using the **no app session** command, use the **(config) app session** command to enable it.

---

For example:

```
# rcmd 192.2.2.2 "show domain" 10
```

## Displaying APP Configurations

To display the APP configuration or session information, use the **show app** command. APP is the method in which private communications links are configured between CSSs in the same content domain. A content domain consists of group of CSSs configured to exchange content information.

The syntax and options for this command are:

- **show app** - Displays whether APP is enabled, its port number, and frame size setting. For example:

```
(config)# show app
```

- **show app session** - Displays all IP session information including the session ID, IP address, and state. For example:

```
(config)# show app session
```

- **show app session ip\_address** - Displays the IP session information including the session ID, IP address, and state. For example:

```
(config)# show app session 192.168.10.10
```

- **show app session verbose** - In addition to displaying the IP session information, the verbose keyword displays detailed information about the IP configuration parameters for the session including the local address, keepalive frequency, authorization and encryption type, frame size, and packet activity. For example:

```
(config)# show app session verbose
```

- **show app session ip\_address verbose** - Displays the same information as the **show app session verbose** command except that it displays information only for the specified IP address. For example:

```
(config)# show app session 192.168.10.10 verbose
```

To display a list of IP addresses, enter **show app ?** or **show app session verbose ?**.

[Table 1-1](#) describes the fields in the **show app** output.

**Table 1-1** Field Descriptions for the show app Command

Field	Description
Enabled or Disabled	Whether all APP sessions are enabled or disabled.
PortNumber	The TCP port number that listens for APP connections. The port can be a number from 1 to 65535. The default is 5001.
MaxFrameSize	The maximum frame size allowed on an APP channel between CSSs. The maximum frame size is a number from 10240 to 65535. The default is 10240.

[Table 1-2](#) describes the fields in the **show app session** output.

**Table 1-2** Field Descriptions for the show app session Command

Field	Description
App Session Information	DNS-resolved host name as defined through the <b>host</b> command.
Session ID	The unique identifier for the session.
IP Address	The IP address for the peer CSS.

**Table 1-2** *Field Descriptions for the show app session Command (continued)*

Field	Description
State	<p>The current state of the session. The possible states include:</p> <ul style="list-style-type: none"> <li>• <b>APP_SESSION_STOP</b> - Indicates that the session is about to be deleted</li> <li>• <b>APP_SESSION_INIT</b> - Indicates that the session is initializing</li> <li>• <b>APP_SESSION_OPEN</b> - Indicates that the connection to the peer has been made</li> <li>• <b>APP_SESSION_AUTH</b> - Indicates that the authentication is occurring</li> <li>• <b>APP_SESSION_UP</b> - Indicates that the session is up</li> <li>• <b>APP_SESSION_DOWN</b> - Indicates that the session is down</li> </ul>
Local Address	The local interface address. If the session is down, no address is displayed.
rcmdEnable	The setting for the sending of remote CLI commands to the peer through the <b>rcmd</b> command. The Enabled setting allows the sending of CLI commands. The Disabled setting disallows the sending of CLI commands. The default setting is enabled.
KalFreq	The time in seconds between sending keepalive messages to this peer CSS. The time can be from 14 to 255 seconds (15 minutes). The default is 14.
Auth Type	The authentication method for the session. The method is either authChallenge for Challenge Handshake Authentication Protocol (CHAP) method or none for no authentication method. The default is no authentication.
Encrypt Type	The encryption method for the packets. The method is either encryptMd5hash for MD5 base hashing method or none for no encryption method. The default is no encryption.

**Table 1-2** *Field Descriptions for the show app session Command (continued)*

Field	Description
MaxFrameSz	The maximum frame size allowed on an APP channel between CSSs. The frame size is a number from 10240 to 65535. The default is 10240.
Pkts Tx	The number of packets sent during the session.
Pkts Rx	The number of packets received during the session.
Pkts Rej	The number of packets rejected during the session.
Last UP event	The day and time of the most recent UP event.
Last DOWN event	The day and time of the most recent DOWN event.
FSM Events	Finite State Machine events as related to the state field.
STOP	The number of APP_SESSION_STOP events. This field will always be at 0.
INIT	The number of APP_SESSION_INIT events.
OPEN	The number of APP_SESSION_OPEN events.
AUTH	The number of APP_SESSION_AUTH events.
UP	The number of APP_SESSION_UP events.
DOWN	The number of APP_SESSION_DOWN events.
Attached Apl	The application identifier.

# Configuring a CSS as an Authoritative DNS Server

Use the **dns-server** command and its options to enable DNS server functionality on a CSS. The options for this global configuration mode command are:

- **dns-server** - Enables the DNS server functionality on a CSS.
- **dns-server zone** - Enables a Proximity Domain Name Server (PDNS) in a Network Proximity configuration, or enables zone-based DNS in a non-proximity configuration.
- **dns-server bufferCount** - Modifies the DNS response buffer count.
- **dns-server respTasks** - Modifies the DNS responder task count.
- **dns-server forwarder** - Enables a DNS server forwarder (a CSS or a fully-functional BIND server), which resolves DNS requests that a CSS cannot resolve.

## Enabling a DNS Server

Use the **dns-server** command to enable the DNS server functionality on a CSS. The syntax of this global configuration mode command is:

```
dns-server
```

For example:

```
(config)# dns-server
```

To disable DNS server functionality on a CSS, enter:

```
(config)# no dns-server
```

## Configuring DNS Server Zones

Use the **dns-server zone** command to enable zone-based DNS on a CSS in a global server load-balancing (GSLB) environment. In a Network Proximity configuration, use this command to enable a PDNS. For more information on Network Proximity, refer to [Chapter 5, Configuring Network Proximity](#).



### Note

Before you enable a Proximity Domain Name Server (PDNS), you must configure APP-UDP and APP. Zone-based DNS also requires APP. For details on configuring APP-UDP, refer to [Chapter 5, Configuring Network Proximity](#) in the section “[Configuring APP-UDP and APP](#)”. For details on configuring APP, see the “[Configuring the Application Peering Protocol](#)” section earlier in this chapter.

The syntax for this global configuration mode command is:

```
dns-server zone zoneIndex {tier1|tier2 {"description"
  {weightedrr|srcip|leastloaded|preferlocal|roundrobin|ip_address
  {weightedrr|srcip|leastloaded|preferlocal|roundrobin } } }
```

The **dns-server zone** command supports the following variables and options:

- *zone\_index* - The numerical identifier of the DNS server zone. The *zone\_index* value must be a unique zone number on the network. In a Network Proximity configuration, this number must match the zone index configured on the Proximity Database (PDB). Enter an integer from 0 to 15. Valid entries are 0 to 5 for tier 1 and 0 to 15 for tier 2. There is no default.
- **tier1**|**tier2** - The optional maximum number of zones (peers) that may participate in the CSS peer mesh. The tier you select must be the same as the tier for the other CSSs participating in the peer mesh. Enter **tier1** for a maximum of 6 zones. Enter **tier2** for a maximum of 16 zones. The default is tier1.
- *description* - Optional quoted text description of the DNS server zone. Enter a quoted text string with a maximum of 20 characters.

- **weightedrr|srcip|leastloaded|preferlocal|roundrobin** - The optional load-balancing method that the DNS server uses to select returned records when a PDB is unavailable or not configured.
  - **weightedrr** - The CSS gives a zone priority over other zones in a peer mesh according to the assigned domain weights. Each CSS in a mesh maintains an internal list of zones ordered from highest to lowest according to weight. The heaviest zone (the zone with the highest weight number) receives DNS requests until it reaches its maximum number of requests, then the next heaviest zone receives DNS requests until it reaches its maximum, and so on. When all the zones have reached their maximum number of requests, the CSS resets the counters and the cycle starts over again.

When you add a new DNS zone, each CSS adds the new zone to its list by weight. In this case, the CSSs do not reset their hit counters. This process prevents flooding of the heaviest zone every time you add or remove a zone.

For example, a domain with a weight of 10 in the local zone will receive twice as many hits as the same domain with a weight of 5 in another zone. Use the **dns-record** command to assign domain weights. See the “[Configuring Domain Records](#)” section later in this chapter.
  - **srcip** - The CSS uses a source IP address hash to select the zone index to return to the client.
  - **leastloaded** - The CSS reports loads and selects a record from the zone that has the least traffic.
  - **preferlocal** - The CSS returns a record from the local zone whenever possible, using roundrobin when it is not possible.
  - **roundrobin** - The CSS cycles among records available at the different zones. This load-balancing method is the default.
- **ip\_address** - The IP address of the PDB. In a proximity configuration, enter the address in dotted-decimal notation (for example, 172.16.2.2). If you choose the zone capabilities (peer mesh) of a DNS server in a non-proximity environment, do not use this variable.

For example:

```
(config)# dns-server zone 0 tier1 "pdns-usa" weightedrr 5
```

To disable the local DNS zone, enter:

```
(config)# no dns-server zone
```

**Note**

If you need to modify a **dns-server zone** value, you must first disable the DNS server using the **no dns-server** command and then remove the zone using the **no dns-server zone** command. Restore the DNS server zone with the value change, then reenables the DNS server.

## Configuring dns-server bufferCount

To change the DNS response buffer count on the CSS, use the **dns-server bufferCount** command. Enter the number of buffers allocated for query response from 2 to 1000. The default is 50.

Use this command with the **show dns-server** command to tune the CSS only if the CSS experiences buffer depletion during normal operation. If the number of available name server buffers (NS Buffers) displayed by the **show dns-server** command drops below 2, use the **dns-server bufferCount** to increase the buffer count. You can also use the reclaimed buffer count as an indication of buffer depletion. When the supply of available buffers is depleted, the CSS reclaims used buffers.

For example:

```
(config)# dns-server bufferCount 100
```

To set the DNS response buffer count to its default value of 50, enter:

```
(config)# no dns-server bufferCount
```

## Configuring dns-server respTasks

To change the DNS responder task count, use the **dns-server respTasks** command. Enter the number of tasks to handle DNS responses as an integer from 1 to 250. The default is 2.

For example:

```
(config)# dns-server respTasks 3
```

To set the DNS responder task count to its default value of 2, enter:

```
(config)# no dns-server respTasks
```

## Configuring a DNS Server Forwarder

Use the **dns-server forwarder** command to configure a DNS server forwarder on a CSS. If the CSS cannot resolve a DNS request, it sends the request to another DNS server to obtain a suitable response. This server, called a DNS server forwarder, can be a fully functional Berkeley Internet Name Domain (BIND) DNS server or a CSS configured for DNS. The CSS sends to the forwarder DNS requests that:

- Are not resolvable by the CSS
- Contain an unsupported request or record type



### Note

---

For Client Side Accelerator (CSA) configurations, the forwarder must be a BIND DNS server. For details on CSA, refer to [Chapter 4, Configuring a Client-Side Accelerator](#).

---

The forwarder resolves the DNS requests and sends DNS responses to the client transparently through the CSS. To monitor forwarder health, a keepalive mechanism (internal to the CSS) sends queries periodically to the forwarder to validate its state.



### Note

---

You must configure at least one local DNS server zone before configuring a DNS server forwarder. For details on DNS server zones, see the “[Configuring DNS Server Zones](#)” section earlier in this chapter.

---

The syntax for this global configuration mode command is:

```
dns-server forwarder [primary ip_address | secondary ip_address | zero]
```

The variables and options are:

- **primary** - Specifies a DNS server as the first choice forwarder. The CSS sends unresolvable requests to the primary forwarder unless it is unavailable, in which case, it uses the secondary forwarder. When the primary forwarder is available again, the CSS resumes sending requests to the primary forwarder.
- **secondary** - Specifies a DNS server as the second choice forwarder.
- *ip\_address* - Specifies the IP address of the forwarder. Enter the address in dotted-decimal notation (for example, 192.168.11.1).
- **zero** - Resets the statistics of both forwarders on a CSS.

For example:

```
(config)# dns-server forwarder primary 192.168.11.1 secondary  
192.168.11.2
```

To delete the primary forwarder on a CSS, enter:

```
(config)# no dns-server forwarder primary
```

## Displaying DNS Server and Zone Information

To display DNS server configuration and database information, use the **show dns-server** command and the **show zone** command. These commands provide the following options and information:

- **show dns-server** - Displays DNS server configuration information
- **show dns-server dbase** - Displays the DNS database information
- **show dns-server stats** - Displays the DNS database statistics
- **show dns-server forwarder** - Displays DNS server forwarder statistics
- **show zone** - Displays information about a specified DNS server zone or all zones in a peer mesh

## Displaying DNS Server Configuration Information

Use the **show dns-server** command to display information about your DNS server configuration. The syntax for this global configuration mode command is:

```
show dns-server
```

[Table 1-3](#) describes the fields in the **show dns-server** output.

**Table 1-3** *Field Descriptions for the show dns-server Command*

Field	Description
DNS Server Configuration	The enable or disable state of the DNS server function on the CSS. When enabled, the CSS acts as the authoritative name server for the content domain.
ACL Index	The ACL index number applied to the DNS server. If this field is 0, no ACL has been applied.
Responder Task Count	The configured DNS server responder task count. These tasks handle responses to incoming DNS query requests. The default is 2. The range is from 1 to 250.
<b>Name Server Buffers</b>	
Total Count	The configured DNS server buffer count. The responder tasks share the buffers to handle incoming queries. The default is 50.
Current Free Count	The number of buffers available (not queried).
Minimum Free Count	The smallest number of buffers that will be available.
Reclaimed Count	The number of buffers forcibly reclaimed by the DNS server software.
Requests Accepted	The number of DNS queries accepted.
Responses Sent	The number of DNS responses sent.
No Error	The number of queries that the DNS server successfully answered.
Format Error	The number of queries received that had a packet format error.

**Table 1-3** Field Descriptions for the `show dns-server` Command (continued)

Field	Description
Server Failure	The number of times that a referenced name server did not reply to a query.
Name Error	The number of queries received that the DNS server was not able to answer.
Not Implemented	The number of queries received requesting an operation that has not been implemented in the DNS server.
Operation Refused	The number of queries the DNS server received that it refused to answer.
<b>Internal Resolver</b>	
Requests Sent	The number of queries sent to another name server for resolution.
Responses Accepted	The number of replies received from another name server.
<b>Proximity Lookups</b>	
Requests Sent	The number of proximity lookups sent to the PDB.
Responses Accepted	The number of proximity lookups received from the PDB.

**Note**

Proximity lookup information is displayed only when you configure a PDB IP address. For information on configuring a PDB, refer to [Chapter 5, Configuring Network Proximity](#), in the “Configuring a Proximity Database” section.

## Displaying DNS Server Database Statistics

Use the `dns-server dbase` command to display DNS server database statistics. The DNS server database contains DNS names that are configured locally or learned from peers and Time to Live (TTL) information for each DNS name. The syntax for this global configuration mode command is:

```
show dns-server dbase
```

Table 1-4 describes the fields in the **show dns-server dbase** output.

**Table 1-4** *Field Descriptions for the show dns-server dbase Command*

Field	Description
DN	The domain name of the entry.
DNSCB	The address of the DNS control block structure to return a DNS query response for the entry. This address is the location best suited to handle the request.
PROX	The address for the proximity record.



**Note**

When DNSCB and PROX have null values (0x0), these values indicate a host table mapping.

## Displaying DNS Server Domain Statistics

Use the **show dns-server stats** to display DNS server domain statistics. The syntax for this global configuration mode command is:

```
show dns-server stats
```

Table 1-5 describes the fields in the **show dns-server stats** output.

**Table 1-5** *Field Descriptions for the show dns-server stats Command*

Field	Description
DNS Name	The domain name entry
Content Name	Where the domain entry is mapped (A-record, NS-record, or host table), or a content rule name
Location	The IP address associated with the entry
Resolve Local	The number of local resolutions performed for the entry
Remote	The number of remote resolutions performed for the entry

## Displaying DNS Server Forwarder Statistics

Use the **show dns-server forwarder** command to display statistics on the CSS for the DNS server forwarders. The syntax for this global configuration mode command is:

```
show dns-server forwarder
```

[Table 1-6](#) describes the fields in the **show dns-server forwarder** output.

**Table 1-6** *Field Descriptions for the show dns-server forwarder Command*

Field	Description
DNS Server Forwarder Primary	The state of the primary forwarder. The states are: <ul style="list-style-type: none"> <li>• Not Configured</li> <li>• Up</li> <li>• Down</li> </ul>
DNS Server Forwarder Secondary	The state of the secondary forwarder. The states are: <ul style="list-style-type: none"> <li>• Not Configured</li> <li>• Up</li> <li>• Down</li> </ul>
State Changes	The number of times that the forwarder's state changed.
Requests Sent	The total number of requests sent to a particular forwarder.
Responses Accepted	The total number of responses received from a particular forwarder.
<b>Totals:</b>	
Request Sent	The total number of requests sent to forwarders (primary and secondary).
Responses Accepted	The total number of responses received from forwarders (primary and secondary).

## Displaying DNS Server Zones

Use the **show zone** command to display information about communication and the state of the specified DNS server zone or proximity zone, or all zones in a peer mesh.

The syntax for this global configuration command is:

```
show zone {zone {verbose} | local | verbose}
```

The variable and options for this command are:

- *zone* - The zone index of a peer. If you omit this variable, this command displays the states of all proximity zones.
- **local** - Display local zone information. This information includes a count of transmitted and received client packet types, the count of client packets, and a count of transmit errors.
- **verbose** - Display extra information per APP negotiation. This information includes a count of transmitted and received client packet types, the count of client packets, and a count of APP transmit errors.

For example:

```
(config)# show zone
```

To display proximity zones, including a count of transmitted and received client packet types, the count of client packets, and a count of APP transmit errors, enter:

```
(config)# show zone 1 verbose
```

[Table 1-7](#) describes the fields in the **show zone** output.

**Table 1-7** *Field Descriptions for the show zone Command*

Field	Description
Index	The zone index of the peer. The initial value is 255. Once peer communications are established using APP, the value changes to the zone index of the peer. If peer communications cannot be negotiated, the value remains at 255.
Description	Zone description as supplied by the peer from the <b>dns-server zone</b> command.

*Table 1-7 Field Descriptions for the show zone Command (continued)*

Field	Description
IP Address	The IP address of the peer. It corresponds to a locally configured APP session.
State	The state of the peer negotiation, which includes: <ul style="list-style-type: none"> <li>• <b>INIT</b> - Initializing. Waiting for local configuration to complete.</li> <li>• <b>SREQ</b> - A connection request message has been sent to the peer.</li> <li>• <b>RACK</b> - An acknowledgment message has been received from the peer.</li> <li>• <b>SACK</b> - An acknowledgment request has been sent to the peer.</li> <li>• <b>OPEN</b> - Negotiations with the peer have completed successfully and the connection is open.</li> <li>• <b>CLOSED</b> - Negotiations with the peer have failed and the connection is closed.</li> </ul>
State Chgs	The number of times the state has transitioned to OPEN and CLOSED.
UpTime	The amount of time that APP has been in the OPEN state.

## Configuring Domain Records

Peer CSS DNS servers that participate in a zone mesh share domain record information using APP (see the “[Configuring the Application Peering Protocol](#)” section earlier in this chapter). The DNS servers use the resulting database of domain names and their zone index information to make zone-based DNS decisions.

Use the **dns-record** command and its options to create domain records on a CSS configured as a DNS server. This command is not available on a CSS configured as a PDB. The CSS uses the following types of domain records to map a domain name to an IP address or to another DNS server, or to accelerate a domain:

- **a** - A domain record mapped to an IP address
- **ns** - A domain record mapped to a DNS server IP address
- **accel** - An accelerated domain associated with a Client Side Accelerator

## Configuring A-Records

Use the **dns-record a** command to create an address record (A-record) on a CSS that maps the domain name to an IP address. Use the **no** form of this command to delete an A-record.

The syntax for this global configuration mode command is:

```
dns-record a dns_name ip_address {ttl_value {single|multiple
{kal-ap|kal-icmp|kal-none {ip_address2 {threshold
{sticky-enabled|sticky-disabled
{usedefault|weightedrr|srcip|leastloaded|preferlocal|roundrobin|
proximity {weight}}}}}}}}}
```

The **dns-record a** command supports the following variables and options:

- *dns\_name* - The domain name mapped to the address record. Enter the name as a lower case unquoted text string with no spaces and a maximum length of 63 characters.
- *ip\_address* - IP address bound to the domain name within the DNS server zone. Enter the address in dotted-decimal notation (for example 172.16.6.7).
- *ttl\_value* - The optional Time to Live (TTL) value in seconds. This value determines how long the DNS client remembers the IP address response to the query. Enter a value between 0 to 65535. The default is 0.
- **single**|**multiple** - Optional number of records to return in a DNS response message. By default, the DNS server returns a single A-record. Specifying **single** returns one A-record. Specifying **multiple** returns two A-records.




---

**Note** To generate **kal-ap** keepalive messages to query agents for load information, CSSs acting as clients must be running the Enhanced feature set. Lower-level CSSs acting as **kal-ap** agents (data centers or DNS servers) do not require the Enhanced feature set. When the Proximity Domain Name Server (PDNS) is directly attached to a server farm, an internal keepalive is used.

---

- **kal-ap** - Optional keepalive message type keyword that specifies the CSS keepalive message. This is the recommended keepalive message type to obtain load information from remote as well as local services based on domains configured on a single content rule.
- **kal-icmp** (default keepalive) - The optional keepalive message type keyword that specifies ICMP echo (ping). To obtain load information from local services only, use the **add dns record\_name** command in the associated content rule. See the [“Adding a DNS Service to a Content Rule”](#) section.
- **kal-none** - The optional keepalive message type keyword that specifies no keepalive messaging.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-icmp
```

- *ip\_address2* - IP address of the local interface receiving CSS keepalive messages. If you omit this address while the keepalive type is specified, the CSS uses the DNS IP address to complete keepalive messaging.
- *threshold* - The load threshold is used only with the kal-ap CSS keepalive. Typically, the CSS keepalive reports 255 when a service is unavailable. This threshold allows the CSS to interpret lower reported numbers as unavailable. For example, if this parameter has a value of 100, all received load numbers greater than or equal to 100 cause the domain record to become unavailable for DNS decisions. Enter a value from 2 to 254. The default is 254.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-ap
123.45.6.12 100
```

- **sticky-enabled** - Causes a CSS DNS server to attempt to send a sticky response to the client for the specified domain. The CSS makes a decision based on one of the following three scenarios:
  - In a global server load-balancing (GSLB) environment without a global sticky database (GSDB), the CSS selects a server based on the srcip hash (regardless of the default load-balancing method) and the availability of the domain in the zone mesh. The use of the srcip hash ensures that the CSS selects a consistent zone for a given source IP address.
  - In a GSLB environment with a GSDB, the CSS sends a lookup request to the Global Sticky Database for the requesting client’s local DNS server. If the GSDB has an entry in its sticky database for the client’s local DNS server IP address, it returns the appropriate zone index to the CSS. The CSS then returns the associated IP address to the client. Otherwise, the CSS selects a zone based on the default load-balancing method and informs the GSDB about the selected zone.
  - In a Network Proximity environment, the CSS configured as a Proximity Domain Name Server (PDNS) first consults the GSDB. If a sticky database entry exists for the client’s local DNS server IP address, the PDNS sends the appropriate IP address to the client based on the zone index returned by the GSDB. If the GSDB does not contain an entry for the client’s local DNS server IP address, the PDNS consults the Proximity Database (PDB).

If the PDB contains an entry for the client’s local DNS server IP address, the PDNS formulates a response to the client based on the ordered zone index returned by the PDB and keepalive information. The PDNS informs the GSDB about the selected zone (performs a “set” function). If the PDB does not have an entry for the client’s local DNS server IP address or the sticky zone is unavailable, the CSS selects a zone based on its default load-balancing method and informs the GSDB about the selected zone.

**Caution**

If you configure any sticky domains in a particular zone, you must configure all sticky domains participating in the peer mesh in that same zone. Otherwise, the thrashing of the sticky zone index will cause DNS Sticky to fail.

- **sticky-disabled** - Disables DNS Sticky for the specified domain on a CSS. This is the default. For details on configuring DNS Sticky, refer to [Chapter 2, Configuring the DNS Sticky Feature](#).

For example:

```
(config)# dns-record a www.home.com 123.45.6.7 15 single kal-ap
172.16.6.12 100 sticky-enabled
```

- **usedefault** - Returns domain records using the default DNS load-balancing method configured for the zone. See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.
- **weightedrr** - Returns domain records based on the weighted roundrobin load-balancing method. This method uses the *weight* value to determine the zone from which the record should be requested.
- **srcip** - Returns domain records using a source IP address hash. For sticky-enabled domains without a GSDB, the CSS uses the srcip method regardless of the configured balance method.
- **leastloaded** - Returns domain records from the zone with the smallest load.
- **preferlocal** - Returns local domain records whenever possible. If no local record exists, the CSS uses the balance method configured for the zone with the lowest zone index.
- **roundrobin** - Returns domain records by cycling among records available at the different zones to evenly distribute the load.
- **proximity** (the default) - Returns domain records based on proximity information. If a PDB is not configured or is unavailable in a zone, the CSS applies the default balance method for the selected zone for DNS resolution.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-ap
172.16.6.12 100 sticky-enabled leastloaded
```



#### Note

---

For sticky-enabled domains without a GSDB, a CSS uses the srcip method regardless of the configured balance method. For sticky-enabled domains with a GSDB, a CSS uses the configured balance method when the GSDB does not contain an entry for the requested domain.

---

- *weight* - Value assigned to a domain in the local zone to determine how many requests the local zone receives for the specified domain compared with other zones in a peer mesh. A domain with a weight of 10 in the local zone will receive twice as many requests as the same domain in another zone with a weight of 5.

Use this parameter with the weighted roundrobin DNS load-balancing method. (See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.) CSSs configured as authoritative DNS servers in a peer mesh share domain weight with each other. Enter an integer from 1 to 10. The default is 1.

The CSS uses the following guidelines when selecting a DNS load-balancing method on a domain basis:

- If a local record exists, the CSS uses the configured domain balance method to determine local DNS resolutions. This rule applies regardless of the keepalive state of the local record.
- If no local record exists, the CSS uses the balance method configured for the zone with the lowest zone index.

For example, consider the following configuration.

Zone	Domain Record	Balance Method
0	www.test.com	leastloaded
1	www.test.com	roundrobin
2	no local record configured for www.test.com	none configured

With this configuration, you can expect the following behavior:

- DNS resolutions occurring on the Zone 0 and Zone 2 DNS servers will use the leastloaded balance method.
- DNS resolutions occurring on the Zone 1 DNS server will use the roundrobin balance method.



#### Note

If you need to modify an existing A-record parameter, you must first remove the record using the **no dns-record a domain\_name** command. Then recreate the A-record with the parameter change using the **dns-record a** command.

## Configuring NS-Records

Use the **dns-record ns** command to create a name server record (NS-record) on a CSS that maps the domain name to the IP address of a lower-level DNS server. Use the **no** form of this command to delete an NS-record.

The syntax for this global configuration mode command is:

```
dns-record ns dns_name ip_address {ttl_value {single|multiple
{kal-ap|kal-icmp|kal-none {ip_address2 {threshold {default|forwarder
{sticky-enabled|sticky-disabled
{usedefault|weightedrr|srcip|leastloaded|preferlocal|roundrobin|
proximity {weight}}}}}}}}}
```

The **dns-record ns** command supports the following options and variables:

- *dns\_name* - The domain name mapped to the address record. Enter the name as a lowercase unquoted text string with no spaces and a maximum length of 63 characters.
- *ip\_address* - IP address of the DNS server bound to the domain name within the DNS server zone. Enter the address in dotted-decimal notation (for example 123.45.6.8).
- *ttl\_value* - The optional Time to Live (TTL) value in seconds. This value determines how long the DNS client remembers the IP address response to the query. Enter a value between 0 and 65535. The default is 0.
- **single**|**multiple** - The optional number of records to return in a DNS response message. By default, the DNS server returns a single NS-record. Specifying **single** returns one NS-record. Specifying **multiple** returns two NS-records.
- **kal-ap** - The optional keepalive message type keyword that specifies the CSS keepalive message. This is the recommended keepalive message type to obtain load information from remote as well as local services based on domains configured on a single content rule.




---

**Note** To use **kal-ap** proximity keepalive messages, lower-level CSSs acting as either data centers or DNS servers must be running the Enhanced feature set. When the Proximity Domain Name Server (PDNS) is directly attached to a server farm, an internal keepalive is used.

---

- **kal-icmp** (default keepalive) - The optional keepalive message type keyword that specifies ICMP echo (ping).
- **kal-none** - The optional keepalive message type keyword that specifies no keepalive messaging.

For example:

```
(config)# dns-record ns www.work.com 172.16.6.8 15 single kal-icmp
```

- *ip\_address2* - IP address of the local interface receiving CSS keepalive messages.
- *threshold* - The load threshold is used only with the kal-ap CSS keepalive. Typically, the CSS keepalive reports 255 when a service is unavailable. This threshold allows the CSS to interpret lower reported numbers as unavailable. For example, if this parameter has a value of 100, all received load numbers greater than or equal to 100 cause the domain record to become unavailable for DNS decisions. Enter a value from 2 to 254. The default is 254.

For example:

```
(config)# dns-record ns www.home.com 172.16.6.7 15 single kal-ap  
123.45.6.12 100
```

- **default** - In a proximity configuration, the CSS uses PDB information to return the next most proximate location. When a PDB is not available or not configured, the CSS uses the roundrobin load-balancing method. There is no failover scenario.
- **forwarder** - Use this option to eliminate a potential single point of failure by providing up to two alternative DNS servers called forwarders. A forwarder can be a CSS configured as a DNS server or a fully-functional BIND DNS server. If an optimal miss occurs (the lower-level DNS server indicated in the NS-record is Down), the PDNS sends the DNS request to the primary or secondary forwarder, depending on forwarder health and configuration. An optimal miss occurs when the PDNS cannot return the NS-record for the zone that the PDB indicated was most proximate.

For this failover to occur, the local NS-record must be in the Down state, and the PDB has indicated the local zone to be the zone most proximate to the client. For information on configuring a DNS server forwarder, see the [“Configuring a DNS Server Forwarder”](#) section earlier in this chapter.

- **sticky-enabled** - Causes the CSS DNS server to attempt to send a sticky response to the client for the specified domain. The CSS makes a decision based on one of the following three scenarios:
  - In a global server load-balancing (GSLB) environment without a global sticky database (GSDB), the CSS selects a server based on the srcip hash (regardless of the default load-balancing method) and the availability of the domain in the zone mesh. The use of the srcip hash ensures that the CSS selects a consistent zone for a given source IP address.
  - In a GSLB environment with a GSDB, the CSS sends a lookup request to the Global Sticky Database for the domain requested by the client. If the GSD has an entry in its sticky database for the client's local DNS server IP address, it returns the appropriate zone index to the CSS. The CSS then returns the associated IP address to the client. Otherwise, the CSS selects a zone based on the default load-balancing method and informs the GSDB about the selected zone.
  - In a Network Proximity environment, the CSS configured as a Proximity Domain Name Server (PDNS) first consults the GSDB. If a sticky database entry exists for the client's local DNS server IP address, the PDNS sends the appropriate IP address to the client based on the zone index returned by the GSDB. If the GSDB does not contain an entry for the client's local DNS server IP address, the PDNS consults the Proximity Database (PDB).

If the PDB contains an entry for the client's local DNS server IP address, the PDNS formulates a response to the client based on the ordered zone index returned by the PDB and keepalive information. The PDNS informs the GSDB about the selected zone. If the PDB does not have an entry for the client's local DNS server IP address or the sticky zone is unavailable, the CSS selects a zone based on its default load-balancing method and informs the GSDB about the selected zone.

**Note**

If you configure any sticky domains in a particular zone, you must configure all sticky domains participating in the peer mesh in that same zone. Otherwise, the thrashing of the sticky zone index will cause DNS Sticky to fail.

- **sticky-disabled** - Disables DNS Sticky for the specified domain. This is the default.

For example:

```
(config)# dns-record ns www.home.com 172.16.6.7 15 single kal-icmp
123.45.6.12 100 sticky-enabled
```



Note

---

For details on configuring DNS Sticky, refer to [Chapter 2, Configuring the DNS Sticky Feature](#).

---

- **usedefault** - The CSS returns domain records using the default DNS load-balancing method configured for the zone. See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.
- **weightedrr** - Returns domain records based on the weighted roundrobin load-balancing method. This method uses the *weight* value to determine the zone from which the record should be requested.
- **srcip** - The CSS returns domain records using a source IP address hash. For sticky-enabled domains without a GSDB, the CSS uses the srcip method regardless of the configured balance method.
- **leastloaded** - The CSS returns domain records from the zone with the smallest load.
- **preferlocal** - The CSS returns local domain records whenever possible. If no local record exists, the CSS uses the balance method configured for the zone with the lowest zone index.
- **roundrobin** - The CSS returns domain records by cycling among records available at the different zones to evenly distribute the load.
- **proximity** (the default) - Returns domain records based on proximity information. If a Proximity Database (PDB) is not configured or is unavailable in a zone, the CSS applies the default balance method for the selected zone for DNS resolution.

For example:

```
(config)# dns-record ns www.home.com 172.16.6.7 15 single kal-icmp
172.16.6.12 100 sticky-enabled leastloaded
```




---

**Note** For sticky-enabled domains without a GSDB, a CSS uses the scrip method regardless of the configured balance method. For sticky-enabled domains with a GSDB, a CSS uses the configured balance method when the GSDB does not contain an entry for the requested domain.

---

- *weight* - Value assigned to a domain in the local zone to determine how many requests the local zone receives for the specified domain compared with other zones in a peer mesh. A domain with a weight of 10 in the local zone will receive twice as many requests as the same domain in another zone with a weight of 5.

Use this parameter with the weighted roundrobin DNS load-balancing method. (See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.) CSSs configured as authoritative DNS servers in a peer mesh share domain weights with each other. Enter an integer from 1 to 10. The default is 1.

The CSS uses the following guidelines when selecting a DNS load-balancing method on a domain basis:

- If a local record exists, the CSS uses the configured domain balance method to determine local DNS resolutions. This rule applies regardless of the keepalive state of the local record.
- If no local record exists, the CSS uses the balance method configured in the zone with the lowest zone index.

For example, consider the following configuration.

Zone	Domain Record	Balance Method
0	www.test.com	leastloaded
1	www.test.com	roundrobin
2	no local record configured for www.test.com	none configured

With this configuration, you can expect the following behavior:

- DNS resolutions occurring on the Zone 0 and Zone 2 DNS servers will use the leastloaded balance method.
- DNS resolutions occurring on the Zone 1 DNS server will use the roundrobin balance method.

**Note**

---

If you need to modify an existing NS-record parameter, you must first remove the record using the **no dns-record ns** *domain\_name* command. Then recreate the NS-record with the parameter change using the **dns-record ns** command.

---

## Removing a Domain Record

Use the **no dns-record command** to remove a domain record.

The syntax for this global configuration mode command is:

```
no dns-record dns_name
```

The *dns\_name* variable maps the DNS name to the address record. Enter the name as a case-sensitive unquoted text string with no spaces and a maximum length of 63 characters.

For example:

```
(config)# no dns-record www.home.com
```

## Resetting the DNS Record Statistics

Use the **dns-record zero** command to reset the DNS record statistics displayed by the **show dns-record** command. The syntax for this global configuration mode command is:

```
dns-record zero [a/ns {dns_name}]accel {dns_name}
```

The options and variables for this command are:

- **a/ns** - Resets the statistics to zero for the domain records that are displayed by the **show dns-record statistics** command (see the “[Displaying DNS-Record Statistics](#)” section later in this chapter) and the **show dns-record proximity** command (refer to [Chapter 5, Configuring Network Proximity](#)).
- *dns\_name* - Resets the statistics for the specified domain name mapped to the DNS record. To view a list of domain names, enter:

```
dns-record zero [a/ns|accel] ?
```

- **accel** - Resets the counters to zero for the accelerated records that are displayed by the **show dns-record accel** command. Refer to [Chapter 4, Configuring a Client-Side Accelerator](#), in the “[Displaying Domain Acceleration Records Statistics](#)” section.

## Displaying DNS Record Information

Use the **show dns-record** command to display statistics about the DNS records that were manually configured or learned from peers.

## Displaying DNS-Record Statistics

Use the **show dns-record statistics** command to display statistics associated with the address records (A-records), name server records (NS-records), or accelerated domain records (accel) configured locally and learned by the CSS from its peers.

The syntax for this global configuration mode command is:

```
show dns-record statistics {dns_name}
```

You may enter an optional domain name target to display content. If you omit the domain name, all domains appear.

For example:

```
(config)# show dns-record statistics
```

Table 1-8 describes the fields in the **show dns-record statistics** output.

**Table 1-8 Field Descriptions for the show dns-record statistics Command**

Field	Description
<Domain name>	Domain name for the record.
Local	State of the local entry for the record. Up indicates that the entry is configured. A “-” character indicates that the entry is learned and not configured. Down indicates that the keepalive failed.
Zone Count	Number of zones where this record is configured.
Zone	Index number for the zone. A “*” character prepending the zone number indicates that the zone is a local entry.
Description	Zone description.
Type	DNS record type: <ul style="list-style-type: none"> <li>• <b>A</b> - Address record</li> <li>• <b>NS</b> - Name-server record</li> <li>• <b>Accel</b> - An accelerated domain associated with a Client Side Accelerator (CSA)</li> </ul>
IP Address	Configured IP address for the zone.
TTL	Time to Live, which indicates how long the receiver of a DNS reply for the given domain should cache the address information. By default, the TTL value is 0, indicating that the name server receiving the response should not cache the information.
Hits	Total number of DNS hits.

## Displaying DNS Record Keepalive Information

Use the **show dns-record keepalive** command to display DNS record keepalive information. The syntax for this global configuration mode command is:

```
show dns-record keepalive {dns-name}
```

The variable for this command is *dns-name*, the domain name associated with the DNS record. You can enter an optional domain name target to display content. If you omit this variable, all DNS records appear.

Table 1-9 describes the fields in the **show dns-record keepalive** output.

**Table 1-9 Field Descriptions for the show dns-record keepalive Command**

Field	Description
Name	Domain name for the record.
Type	Keepalive message type for the record: Accel, AP, ICMP, or none.
IP	Destination IP address of the keepalive message.
State	State of the record, either UP or DOWN.
Transitions	Number of state transitions.
Load	Load for the record, which applies only to an AP record type. All other types always have a load of “-”, indicating an undetermined load (load reports are not being received).  If the load value exceeds the threshold value, the DNS server removes the DNS record from eligibility.
Threshold	Configured load threshold for the record. This threshold applies only to an AP record type. Record types of ICMP and none do not use the threshold value.

## Displaying the DNS Record Weight

Use the **show dns-record weight** command to display the configured weight and the number of hits for all domains or the specified domain. The syntax for this global configuration command is:

```
show dns-record weight {dns_name}
```

The *dns-name* variable for this command is the domain name associated with the DNS record. You can enter an optional domain name target to display information for the specified domain record. If you omit this variable, all DNS records appear.

Table 1-10 describes the fields in the **show dns-record weight** output.

**Table 1-10** *Field Descriptions for the show dns-record weight Command*

Field	Description
Name	Domain name for the record.
Total Hits	Total number of hits in all zones for the specified domain name.
Zone	Zone index for each zone where the domain record resides. An asterisk indicates the local zone.
Description	Text description of the zone.
IP Address	IP address of the DNS server bound to the domain name within the DNS server zone.
Weight	Configured weight value for the record.
Current Hits	Current number of hits for the domain record in the zone.
Total Hits	Total number of hits for the domain record in the zone.

## Configuring DNS Using Content Rules

The following sections describe how to configure DNS using content rules and associated commands.



### Note

The recommended method for configuring DNS in a global server load balancing environment is zone-based DNS. For details on enabling DNS server functionality on a CSS, see the [“Configuring a CSS as an Authoritative DNS Server”](#) section earlier in this chapter.

## Configuring CSS DNS Peering

Use the **dns-peer** command and its options to enable DNS peer functionality on a CSS. Peer functionality includes the sharing of content rules. The syntax and options for this global configuration mode command are:

- **dns-peer interval** - Sets the time between the load reports to the CSS DNS peers
- **dns-peer receive-slots** - Sets the maximum number of DNS names that the CSS can receive from each CSS DNS peer
- **dns-peer send-slots** - Sets the maximum number of DNS names that the CSS can send to each CSS DNS peer

## Configuring the DNS Peer Interval

To set the time between generating load reports to the CSS DNS peers, use the **dns-peer interval** command. Enter the peer interval time from 5 to 120 seconds. The default is 5.

For example:

```
(config)# dns-peer interval 60
```

To reset the DNS peer interval to its default value of 5 seconds, enter:

```
(config)# no dns-peer interval
```

## Configuring DNS Peer Receive Slots

To set the maximum number of DNS names that the CSS can *receive* from each CSS DNS peer, use the **dns-peer receive-slots** command. Enter a number from 128 to 1024. The default is 128. Use this command to tune a heavily-accessed CSS that is resolving more than 128 DNS names.

For example:

```
(config)# dns-peer receive-slots 200
```

To reset the DNS peer receive slots number to its default of 128, enter:

```
(config)# no dns-peer receive-slots
```

## Configuring DNS Peer Send Slots

To set the maximum number of DNS names that the CSS can *send* to each CSS DNS peer, use the **dns-peer send-slots** command. Enter a number from 128 to 1024. The default is 128. Use this command to tune a CSS that is reporting over 128 DNS names to the peer.

For example:

```
(config)# dns-peer send-slots 200
```

To reset the DNS peer send slots number to its default of 128, enter:

```
(config)# no dns-peer send-slots
```

## Displaying DNS Peer Information

To display the DNS peering configuration, use the **show dns-peer** command.

For example:

```
(config)# show dns-peer
```

[Table 1-11](#) describes the fields in the **show dns-peer** output.

**Table 1-11** *Field Descriptions for the show dns-peer Command*

Field	Description
CSD Peer Rcv Slots	The configured maximum number of DNS names that the CSS can receive from each CSS DNS peer over an APP connection. The default is 128. The range is from 128 to 1024.
CSD Peer Snd Slots	The configured maximum DNS names that the CSS can send to each CSS DNS peer. The default is 128. The range is from 128 to 1024.
Peer Report Interval	The configured time in seconds between sending load reports to CSS DNS peers over an APP connection. The default is 5. The range is from 5 to 120.

## Configuring the DNS Exchange Policy for an Owner

To set the DNS exchange policy for an owner, use the **dns** command. The syntax and options for this owner mode command are:

- **no dns** - Sets no DNS exchange policy for this owner (default). This owner is hidden from the CSS peer.
- **dns accept** - Accepts all content rules for this owner proposed by the CSS peer.
- **dns push** - Advertises the owner and push all its content rules to the CSS peer.
- **dns both** - Advertises the owner and push all its content rules to the CSS peer, and accept all this owner's content rules proposed by the CSS peer.

For example:

```
(config-owner[arrowpoint])# dns both
```

## Adding a DNS Service to a Content Rule

To specify a DNS name that maps to a content rule, use the **add dns** command. Enter the DNS name as a lowercase unquoted text string with no spaces and a length of 1 to 31 characters.



### Note

---

The **add dns** command is part of the CSS Standard feature set.

---

When you add the DNS name to the content rule, you may also enter an optional Time to Live (TTL) value in seconds. This value specifies how long the DNS client remembers the IP address response to the query. Enter a value from 0 to 255. The default is 0.



### Note

---

You must configure the TTL when you add the DNS name to the content rule. To add a TTL to an existing rule, use the **remove dns** command to remove the dns name. Then use the **add dns** command to reconfigure the DNS name with a TTL value.

---

For example:

```
(config-owner-content[arrowpoint-rule1])# add dns
www.arrowpoint.com 36
```

## Removing DNS from a Content Rule

To remove a DNS name from a content rule, use the **remove dns** command with the DNS name you wish to remove. Enter the DNS name as a case-sensitive unquoted text string with no spaces and a maximum of 31 characters.

**Note**

The **remove dns** command is part of the CSS Standard feature set.

For example:

```
(config-owner-content [arrowpoint-rule1])# remove dns  
www.arrowpoint.com
```

To display a list of DNS names, enter:

```
(config-owner-content [arrowpoint-rule1])# remove dns ?
```

## Configuring Source Groups to Allow Servers to Resolve Domain Names Using the Internet

The CSS provides support to enable servers to resolve domain names using the Internet. If you are using private IP addresses for your servers and wish to have the servers resolve domain names using domain name servers that are located on the Internet, you must configure a content rule and source group. The content rule and source group are required to specify a public Internet-routable IP address (Virtual IP address) for the servers to allow them to resolve domain names.

To configure a server to resolve domain names:

1. Configure the server, if you have not already done so.

The following example creates *Server1* and configures it with a private IP address 10.0.3.251 and activates it.

```
(config)# service Server1  
(config-service[Server1])# ip address 10.0.3.251  
(config-service[Server1])# active
```

2. Create a content rule to process DNS replies. This content rule is in addition to the content rules you created to process Web traffic. The content rule enables the CSS to perform Network Address Translation (NAT) to translate inbound DNS replies from the public VIP address (192.200.200.200) to the server's private IP address (10.0.3.251).

The following example creates content rule *dns1* with a public Virtual IP address (VIP) 192.200.200.200 and adds server *Server1*.

```
(config-owner[arrowpoint])# content dns1
(config-owner-content [arrowpoint-dns1])# vip address
192.200.200.200
(config-owner-content [arrowpoint-dns1])# add service Server1
(config-owner-content [arrowpoint-dns1])# active
```

3. Create a source group to process DNS requests. The source group enables the CSS to NAT outbound traffic source IP addresses from the server's private IP address (10.0.3.251) to the public VIP address (192.200.200.200).

To prevent server source port collisions, the CSS NATs the server's source IP address and port by translating the:

- Source IP address to the IP address defined in the source group.
- Port to the port selected by the source group. The source group assigns each server a unique port for a DNS query so that the CSS can match the DNS reply with the assigned port. This port mapping enables the CSS to direct the DNS reply to the correct server.

The following example creates source group *dns1* with public VIP address 192.200.200.200 and adds server *Server1*.

```
(config)# group dns1
(config-group[dns1])# vip address 192.200.200.200
(config-group[dns1])# add service Server1
(config-group[dns1])# active
```

# Displaying Domain Summary Information

To display content domain summary information, use the **show domain** command. The syntax and options are listed below. For options that require an IP address, specify the IP address for the peer.

- **show domain** - Displays content domain summary information including the number of domain peers and information about each peer.
- **show domain ip\_address send|receive** - Displays content domain summary information including the number of domain peers and information for the specified peer IP address. To see a list of addresses, enter **show domain ?**.
  - Include the **send** option to display only the send load reports and transmit message statistics.
  - Include the **receive** option to display only the receive load reports and receive message statistics.
- **show domain hotlist** - Displays configuration information about domain hotlists.
- **show domain owners** - Displays shared owner names.
- **show domain owners ip\_address** - Displays shared owner names for the specified peer IP address.
- **show domain rules** - Displays locally created or negotiated names.
- **show domain rules ip\_address** - Displays locally created or negotiated names for the specified peer IP address.

Table 1-12 describes the fields in the **show domain** output.

**Table 1-12 Field Descriptions for the show domain Command**

Field	Description
Content Domain Summary	The number of domain peers.
Peer	The address for the peer.
CCC State	The state of the master FSM (finite state machine) that negotiates the CAPP (CCC) link.
OWN State	The state of the owner policy negotiation FSM that determines the owners about whom the peers will share domain name and rule information.
Rule State	The state of the rule policy negotiation FSM that exchanges individual domain name and rule matching criteria and load report information.
SendSlots	The number of individual domain name rules on which the CSS will send load reports to the peer.
ReceiveSlots	The number of individual domain name rules on which the CSS will receive load reports from the peer.
Interval	The time interval in seconds that load reports are sent to the peer.
MinRespTime	The minimum local flow response time. This number is shared with the peer to be used in conjunction with load numbers to normalize the load numbers shared between peers.
MaxRespTime	The maximum local flow response time. This number is shared with the peer to be used in conjunction with load numbers to normalize the load numbers shared between peers.
Policy	The negotiated load report send and receive policies.
Sending Load Reports for	The list of domain names for which the CSS is sending load reports to the peer.

*Table 1-12 Field Descriptions for the show domain Command (continued)*

Field	Description
Receiving Load Reports for	The list of domain names for which the CSS is receiving load reports from the peer.
CCC Msg stats	The number of times each of the message types used in the CCC/OWN/Rule FSM negotiations with the peer has been sent or received.

■ Displaying Domain Summary Information