Configuring the CSS as a Domain Name System Server

This chapter describes how to configure a CSS as a Domain Name System (DNS) server to respond to DNS requests by resolving a domain name to an IP address. Information in this chapter applies to all CSS models, except where noted.

Note The DNS feature requires the CSS Enhanced feature set license.

This chapter provides the following major sections:

- Overview of the CSS DNS Feature
- Overview of the CSS Application Peering Protocol
- Configuring the Application Peering Protocol
- Displaying APP Configurations
- Configuring Zone-Based DNS on a CSS
- Configuring Content Rule-Based DNS on a CSS
- Displaying CSS DNS Information

Overview of the CSS DNS Feature

The CSS DNS feature enables you to configure one or more CSSs to construct highly available, distributed, and load-sensitive websites. Groups of CSSs may host many distributed websites concurrently. These groups make decisions that can be configured independently for each distributed website using local and remote load-balancing and domain 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 domain records (zone-based DNS only), content rules (content rule-based DNS only), and service availability and load (both zone-based and rule-based DNS).

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 configure a CSS as an authoritative DNS server. An authoritative DNS server does not depend on lower-level name servers for the answer to a DNS request. Instead, the CSS responds directly to the request by sending a locally configured or learned address record (A-record) of the subdomain to the resolver (the requestor of the DNS resolution, which could be a client, the client local DNS server, or another DNS server). A domain is a subdomain of another domain if it is contained within that other domain. For example, www.cisco.com is a subdomain of cisco.com.

You can also configure a CSS to query lower-level name servers to help resolve a DNS request. You accomplish this by configuring name server records (NS-records), which point to the lower-level DNS servers, on the CSS.

For example, when a user clicks 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.
2. The local DNS server learns the interface address of the CSS through normal DNS processing.
3. The local client DNS server requests address resolution from the CSS authoritative DNS server.

4. The CSS authoritative DNS server returns the VIP address of the best location (based on server availability and load) where the client can retrieve the content.
5. The local DNS server responds to the client with the VIP.
6. 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 for the subdomain.

You can configure DNS on a CSS in either of the following two ways:

- Zone-based DNS
- Content rule-based DNS

Zone-Based DNS

Zone-based DNS creates content domains in geographical areas or zones using the **dns-server zone** command. For example, each zone may be a country, a state, or a city located in different parts of the world. Each CSS peer acts as an authoritative DNS server for the subdomain or subdomains that it represents in a zone. A CSS uses its locally configured or learned A-records or name server records (NS-records) to resolve DNS requests for configured subdomains. This is the recommended method of configuring global server load balancing (GSLB) on a CSS.

Like content rule-based DNS (see the next section), zone-based DNS uses the Application Peering Protocol (APP) for communication and information exchange between CSS peers. However, in this case, the CSS peers exchange subdomain address records (A-records) or name-server records (NS-records), with load and status information.

When a CSS receives a DNS request from a client local DNS server, the CSS attempts to resolve the domain name based on the local domains that are directly connected to it or its APP peers. If the CSS can resolve the domain name locally, it uses its A-records that were locally configured or learned through APP to resolve the domain name to an IP address. The CSS then responds with the IP address associated with the domain name to the client local DNS server.

If the CSS cannot resolve the domain name using its A-records, it forwards the request to a lower-level name server (may be a CSS) using its locally configured and learned NS-records. The lower-level DNS server returns an A-record to the CSS, which uses the A-record to resolve the domain name and then forwards the IP address to the local DNS server.

For details about configuring zone-based DNS, see to the "Configuring Zone-Based DNS on a CSS" section.

Content Rule-Based DNS

Content rule-based DNS uses domain names configured in content rules to resolve DNS requests. Like zone-based DNS, rule-based DNS uses APP for communication and information exchange between CSS peers. Unlike zone-based CSS peers, rule-based DNS peers exchange owner names and content rules according to their configured exchange policies as well as service load and status information.

For details about configuring rule-based DNS, see to the "Configuring Content Rule-Based DNS on a CSS" section.

Overview of the CSS Application Peering Protocol

CSSs configured within the same zone or content domain initiate communication using 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 service load and domain information. APP provides a guaranteed and private communications channel for this exchange. APP is used by both zone-based and rule-based CSS DNS servers.

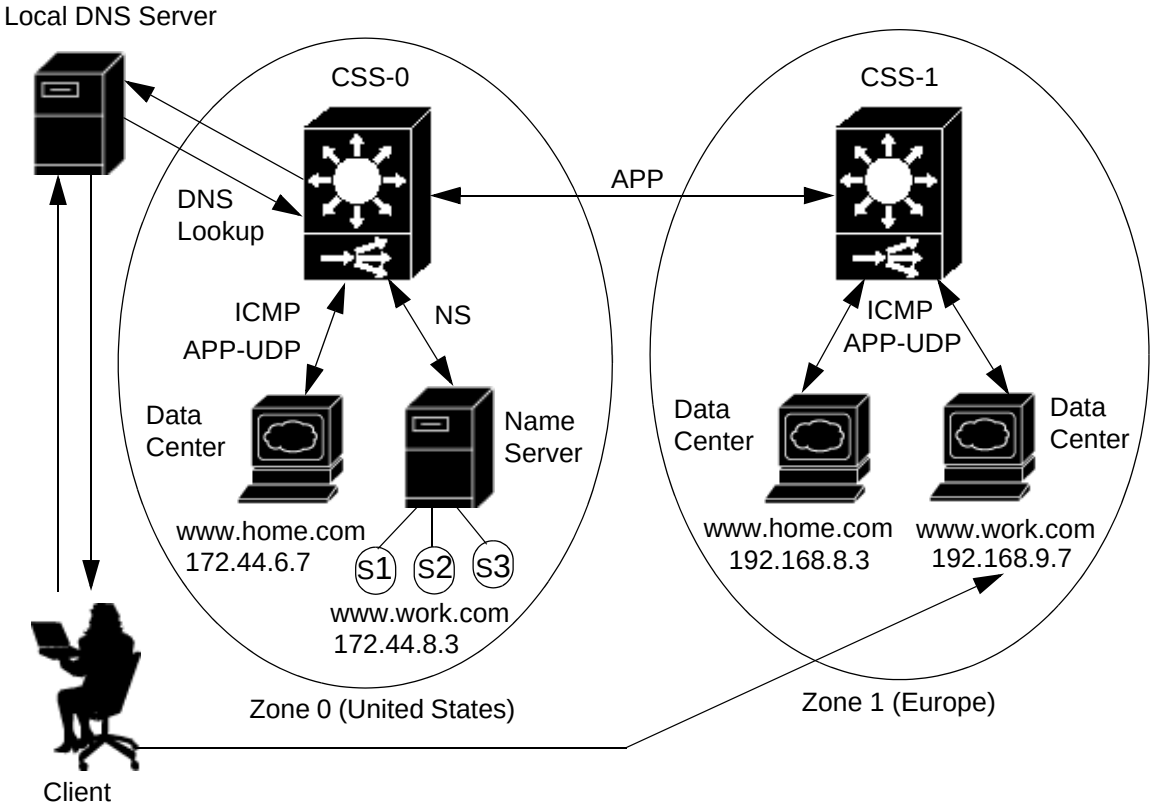
APP and Zone-Based DNS

In addition to exchanging service load and status using APP, CSS peers in a zone-based DNS configuration exchange A-records for directly connected domains and NS-records for lower-level data centers. This domain record exchange allows remote peers to learn about other peer CSS domains.

Figure 1-1 illustrates how zone-based DNS operates with two CSS peers in different zones. For example, when a client requests *www.work.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 authoritative DNS server (in this example, CSS-0) determines server availability based on the DNS balance type configured for the zone or the domain record as follows:
   - If CSS-0 determines that the best selection is a name server (NS) record, the CSS begins a recursive query of the name server to determine an authoritative response.
   - If CSS-0 finds that the best selection is an address record (A-record), it formulates an authoritative response immediately. In this example, CSS-0 decides that the best selection is an A-record (learned through the peer mesh with CSS-1) for a data center in Zone 1.
4. CSS-0 sends an authoritative response that contains the resolved IP address of *www.work.com* to the client local DNS server.
5. The local DNS server notifies the client that sufficient domain name resolution information is available to establish a data connection to *www.work.com*.
6. Lastly, the client uses the local DNS server response information (IP address) to connect to a service and starts receiving content. In this example, the best service is located in Zone 1 at IP address 192.168.9.7.

Figure 1-1 Example of GSLB Using Two Zones



APP and Rule-Based DNS

For rule-based DNS, CSS 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-config 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 authoritative DNS 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.

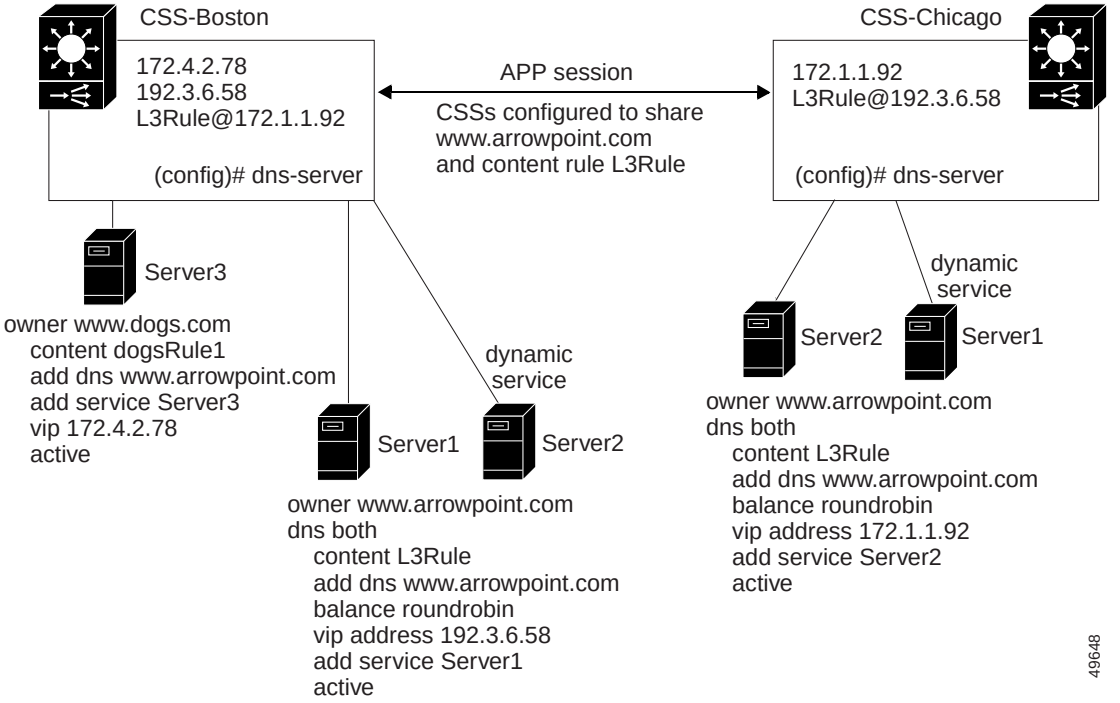
   If the CSS is configured as DNS balance type **dnsbalance roundrobin**, the CSS resolves requests by evenly distributing the load to resolve domain names among local and remote content domain sites.

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

4. The DNS server forwards the resolved VIP to the client.
5. The client uses the VIP to access the content.

Figure 1-2 illustrates two peer CSSs configured as authoritative DNS servers using rule-based DNS. 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 1-2 Example of GSLB (Static Proximity) Using Two CSSs Configured as Authoritative DNS Servers



Configuring the Application Peering Protocol

The Application Peering Protocol feature is part of the CSS Enhanced feature set. You need to configure APP prior to configuring the DNS server for both zone-based and rule-based DNS.

To configure APP, 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 APP
- **app port** - Sets the TCP port that listens for APP connections
- **app session** - Creates an APP session

Enabling APP

To enable APP, use the **app** command on each CSS peer. For example:

```
(config)# app
```

To disable all APP sessions, enter:

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

Configuring the 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 15000 and CSS-B is configured for frame size 20000. Once the session is established, CSS-B will use frame size 15000.

For example:

```
(config)# app framesz 15000
```

To restore the default frame size to 10240, enter:

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

Configuring the 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 1500
```

To restore the default port number to 5001, enter:

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

Note APP frame failures occur in some networks as the result of routing configurations. Suppose the local CSS peer (CSS A) sends an APP frame to a remote CSS peer (CSS B) on a particular interface (B1). CSS B responds from an interface (B2) whose IP address is different from the original destination address. CSS A drops the APP response frame causing APP to fail. This issue is compounded when you use equal-cost multi-path (ECMP) routing. In this case, configure a static route for the APP session.

Configuring an APP Session

To create a content domain between two or more CSSs, the CSSs use APP sessions. An APP session allows the CSSs to share the same content rule, load, and domain information, which are used to determine the best site to resolve a DNS request. To create an APP session between two CSSs, use the **app session** command.

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 and no spaces.
- **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 CLI commands (including playing scripts) to remote CSS peers over an APP session, use the **rcmd** command. Before you use this command, ensure that:

- Circuit addresses are set up on both the local and the remote CSSs
- Routing is set up properly on both CSSs
- APP is configured on both CSSs (see the "Configuring the Application Peering Protocol" section)

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 *host* - 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 (""). If the CLI command itself requires quotes, use single quotes around that part of the command string. For example:

  rcmd 192.168.12.23 "script play test '*argument1 argument2***'"**

  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.

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 you configure private communications links between CSSs in the same content domain. A content domain consists of two or more 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 its 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** - Displays the IP session information. In addition, 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 session ?**.

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	All APP sessions are either 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 **host** 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	The current state of the session. The possible states include: • **APP_SESSION_STOP** - Indicates that the session is about to be deleted • **APP_SESSION_INIT** - Indicates that the session is initializing • **APP_SESSION_OPEN** - Indicates that the connection to the peer has been made • **APP_SESSION_AUTH** - Indicates that authentication is occurring • **APP_SESSION_UP** - Indicates that the session is up • **APP_SESSION_DOWN** - Indicates that the session is down
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 **rcmd** 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 is always zero.
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 Zone-Based DNS on a CSS

Zone-based DNS is the recommended method for configuring global server load balancing (GSLB) on a CSS. GSLB refers to a configuration where two or more geographically distributed CSSs represent one or more domains. Each CSS:

- Acts as an authoritative DNS server for the domain it represents
- Shares domain records and load information with other CSS peers using APP

Each zone is represented by one CSS or a redundant pair of CSSs in the peer mesh. Additional CSSs can exist in the same zone as lower-level DNS servers or content servers. A zone can be a geographic area such as a country, a region, or a city. On each CSS, you configure domain records that the CSS uses to resolve DNS requests. These records can be one of the following types:

- Address record (A-record) - Any domain that represents a data center that is not front-ended by another DNS server and that can be translated to an IP address.
- Name server record (NS-record) - Any domain that is front-ended by a lower-level DNS server (not necessarily a CSS).

This section contains the following topics:

Zone-Based DNS Quick Start

Table 1-3 provides a quick overview of the steps required to configure zone-based DNS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 1-3.

Table 1-3 Zone-Based DNS Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Enable the Application Peering Protocol-User Datagram Protocol (APP-UDP) only if you intend to use the DNS record keepalive type of **kal-ap** or **kal-ap-vip**. See the "Configuring Domain Records" section. `(config)# app-udp`

Table 1-3 Zone-Based DNS Configuration Quick Start (continued)

Task and Command Example
3. Enable APP to allow the local CSS to communicate with remote CSS peers. See the "Configuring the Application Peering Protocol" section. `(config)# app`
4. Configure the local CSS zone. Specify a zone number, tier level, optional text description, and optional default load-balancing method and weight. See the "Configuring DNS Server Zones" section. `(config)# dns-server zone 0 tier1 "usa" weightedrr`
5. Configure an APP session with the remote CSS that is participating in the peer mesh with the local CSS. The IP address you enter is a local interface address on the remote CSS. See the "Configuring the Application Peering Protocol" section. `(config)# app session 172.16.2.5`
6. Create A-records for domains in the local zone. Specify the domain name mapped to the address record and the IP address bound to the domain name. Include an optional time to live (TTL) value, the number of records to return in a DNS response message, and the keepalive message type. See the "Configuring Domain Records" section. `(config)# dns-record a www.home.com 192.168.6.7 0 single kal-ap`
7. Create NS-records for domains on other DNS servers within the local zone. Specify the domain name mapped to a domain IP address. Include an optional TTL value, the number of records to return in a DNS response message, and the keepalive message type. See the "Configuring Domain Records" section. `(config)# dns-record ns www.work.com 172.16.6.8 0 single kal-ap`
8. (Optional) Create content rules for local A-records. In some configurations, there may not be any local content rules or services. For details on creating content rules, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide.*
9. Configure the CSS to act as a DNS server. See the "Enabling a DNS Server" section. `(config)# dns-server`
10. (Optional) Verify the configuration using the **show** commands described in the "Displaying CSS DNS Information" section.

The following running-config example shows the results of entering the commands described in Table 1-3.

```
!************************** GLOBAL ***************************

  app-udp

  dns-server zone 0 tier1 "usa" weightedrr
  dns-record a www.home.com 192.168.6.7 0 single kal-ap
  dns-record ns www.work.com 172.16.6.8 0 single kal-ap
  dns-server

  app
  app session 172.16.2.5
```

Configuring a DNS Server

As a DNS server, a CSS resolves domain names to IP addresses when it receives DNS requests from clients. To configure a CSS as a DNS server, use the **dns-server** command and its options. The options for this global configuration mode command are:

- **dns-server bufferCount** - Modifies the DNS response buffer count.
- **dns-server forwarder** - Enables a DNS forwarder (a CSS or a fully functional BIND server), which resolves DNS requests that a CSS cannot resolve.
- **dns-server respTasks** - Modifies the DNS responder task count.
- **dns-server zone** - Enables zone-based DNS in a non-proximity configuration or enables a Proximity Domain Name Server (PDNS) in a Network Proximity configuration. For details about Network Proximity, see Chapter 5, Configuring Network Proximity.
- **dns-server zone load** - Specifies how the CSS handles the least-loaded load-balancing method.

Note You need to configure APP before you configure the DNS server as shown in Table 1-3.

Enabling a DNS Server

To enable a CSS to resolve domain names to IP addresses, use the **dns-server** command.

Note The **dns-server** command is part of the CSS Enhanced feature set and requires a separate license key.

For example:

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

To disable DNS server functionality on a CSS, enter:

```
(config)# no 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 (see Table 1-5) to tune the CSS only if the CSS experiences buffer depletion during normal operation. If the number of available name server buffers (NS Buffers) in the Current Free Count field drops below 2, use the **dns-server bufferCount** command to increase the buffer count. You can also use the Minimum Free Count (a low-water mark indicator) and the Reclaimed Buffer Count fields as indications 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 a DNS Forwarder

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 forwarder, can be any 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

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 forwarder. For details on DNS server zones, see the "Configuring DNS Server Zones" section earlier in this chapter.

To configure a DNS forwarder on a CSS, use the **dns-server forwarder** command. For Client Side Accelerator (CSA) configurations, the forwarder must be a full-featured DNS server. For details on CSA, see Chapter 4, Configuring a Client-Side Accelerator.

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
```

Configuring the Number of DNS Responder Tasks

The responder task is the part of the CSS DNS server that responds to DNS requests from clients. Typically, the default number of 2 responder tasks is sufficient to handle the volume of DNS requests received by the CSS. 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 DNS Server Zones

To enable zone-based DNS on a CSS in a global server load-balancing (GSLB) environment, use the **dns-server zone** command and its options. In a Network Proximity configuration, use this command to enable a PDNS. For more information on Network Proximity, see Chapter 5, Configuring Network Proximity.

Note Before you enable a Proximity Domain Name Server (PDNS) or Zone-based DNS, you must configure APP. If you are using Network Proximity or the **kal-ap** keepalive type, you must also configure APP-UDP. For details on configuring APP, see the "Configuring the Application Peering Protocol" section earlier in this chapter. For details on configuring APP-UDP, see the "Configuring APP-UDP and APP" section in Chapter 5, Configuring Network Proximity.

The syntax for this global configuration mode command is:

dns-server zone *zone_index* {**tier1**|**tier2** {**"***description***"** {**weightedrr**|**srcip**|**leastloaded**|**preferlocal**|**roundrobin**|*ip_address* {**weightedrr**|**srcip**|**leastloaded**|**preferlocal**|**roundrobin**}{*weight*}}}}

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 default load-balancing method that the DNS server uses to select returned records when a PDB is unavailable or not configured. The load-balancing method you select here can be overridden by the load-balancing method that you assign to individual DNS records.
  - **weightedrr** - Use this load-balancing method with the *weight* variable to define the default weight applied to all DNS records in the local zone. 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* value for a particular domain) 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 zone with a domain that has a weight of 10 receives twice as many hits as a zone with the same domain configured with a weight of 5. 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.
- *weight* - If you do not configure a weight for individual records using the **dns-record** command, then use this variable with the **weightedrr** load-balancing option to define the default weight applied to all DNS records in the local zone. Enter an integer from 0 to 10. The default is 1. To display the weight that you configured on a record using either the **dns-server zone** command or the **dns-record** command, enter the **show dns-record weight** command.

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, and then reenable the DNS server.

Configuring dns-server zone load

Use the **dns-server zone load** command to configure how the CSS handles the least-loaded balance method.

The syntax for this global configuration mode command is:

dns-server zone load [**reporting**|**frequency** *seconds*|**variance** *number*]

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

- **reporting** - Enables the processing of local DNS server zone load information and the sharing of it with peers. The default is enabled.
- **frequency** - Specifies the period of time between the processing of local DNS server load information and the subsequent delivery of load information to peers.
- *seconds* - Specifies the frequency time (in seconds). Enter an integer between 5 and 300 seconds (5 minutes). The default is 30 seconds.
- **variance** - Specifies the range of load numbers between zones that are considered similar for the least-loaded algorithm. If the load numbers of all zones are within the specified range, the CSS uses minimum response times to identify the least-loaded site.

  Note For GSLB, we recommend that you set the same load variance value on all CSSs in a peer mesh. If you configure the absolute load calculation method, we recommend that you configure a load variance of 0. For information about the absolute load calculation method, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

- *number* - Specifies the variance value. Enter an integer between 0 and 255. The default is 255.

For example, to set the DNS server zone load frequency to 120, enter:

```
(config)# dns-server zone load frequency 120
```

To disable DNS server zone load reporting, enter:

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

Note To enable or disable **dns-server zone load reporting**, you must first disable the DNS server using the **no dns-server** command. Then issue the **dns-server zone load reporting** or the **no dns-server zone load reporting** command.

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.

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:

- address record (A-record) - Domain record mapped to an IP address.
- name server record (NS-record) - Domain record mapped to a DNS server IP address.
- accelerated record - Domain record associated with a Client Side Accelerator. See Chapter 4, Configuring a Client-Side Accelerator.

To create an address record (A-record) on a CSS that maps the domain name to an IP address or to create a name server record (NS-record) that maps the domain name to the IP address of a lower-level DNS server, use the **dns-record** command. Use the **no** form of this command to delete an A-record or an NS-record. This command is not available on a CSS configured as a PDB.

The syntax for this global configuration mode command is:

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

dns-record ns *dns_name ip_address* {*ttl_value* {**single**|**multiple** {**kal-ap-vip**|**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** command supports the following variables and options:

- **a** - Specifies that the CSS generates the DNS response based on local information. This option creates an A-record on the CSS that maps the domain name to an IP address.
- **ns** - Specifies that the CSS passes the request to another DNS server whose IP address is contained in a NS-record. This option creates a name server record (NS-record) on a CSS that maps the domain name to the IP address of a lower-level DNS server.The CSS exhibits a pass-through behavior with respect to NS-records. That is, the CSS that receives the original DNS request queries the device specified by the IP address in the NS-record.
- *dns_name* - The fully qualified 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 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.
- **kal-ap-vip** - Optional Cisco CSS keepalive message type keyword used by a CSS client to request load information for the VIP specified in the *ip_address* value from the CSS agent specified in the *ip_address2* value. Use this option to allow a CSS client to query a local or remote CSS agent for load information for a VIP configured on one or more content rules. The keepalive frequency is set to 6 seconds and is not configurable. For details on configuring **kal-ap-vip**, see the "Configuring kal-ap-vip" section.

Note To generate **kal-ap** or **kal-ap-vip** 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** or **kal-ap-vip** 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. Use this option to obtain load information from remote as well as local services based on domains configured on a single content rule. If you configure **kal-ap** on a CSS acting as a client, you must also configure the **add dns** command in a content rule with the appropriate domain names on the CSS acting as an agent. The agent responds with the load information for the configured domain names. The keepalive frequency is set to 6 seconds and is not configurable. For details about the **add dns** command, see the "Adding a DNS Name to a Content Rule" section.
- **kal-icmp** (Default keepalive) - The optional keepalive message type keyword that specifies ICMP echo (ping). The keepalive frequency is set to 6 seconds and is not configurable.
- **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. Generally, this is required for kal-ap or kal-ap-vip.
- *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
```

- **default** - For NS-records only. In a Network 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** - For NS-records only. 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 any DNS server including a CSS configured as a DNS server. If the lower-level DNS server indicated in the NS-record is Down, the CSS sends the DNS request to the primary or secondary forwarder if configured. For information on configuring a DNS forwarder, see the "Configuring a DNS Forwarder" section.

  In a Network Proximity configuration, if an optimal miss occurs (the lower-level DNS server that was 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.
- **sticky-enabled** - Causes a CSS DNS server to attempt to send a sticky response to the client for the specified domain. For details on configuring DNS Sticky, see Chapter 2, Configuring the DNS Sticky Feature. 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 causes 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, see 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. Requires the kal-ap keepalive for remote sites or ICMP keepalives for local content rules.
- **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.

Note The above domain record load-balancing methods override the load-balancing options configured with the **dns-server zone** command. We recommend that you do not mix domain record load-balancing methods for the same domain.

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 hits the local zone receives for the specified domain compared with other zones in a peer mesh. For example, a zone with a domain weight of 10 receives twice as many hits as another zone with the same domain configured 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 through APP. Enter an integer from 0 to 10. The default is 1. The weight value that you assign to an individual domain record takes precedence over the default record weight that you configure for the local zone.

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 use the least-loaded balance method.
- DNS resolutions occurring on the Zone 1 DNS server 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.

Removing a Domain Record

To remove a domain record from the running-config, use the **no dns-record** command.

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

To reset the DNS record statistics displayed by the **show dns-record** command, use the **dns-record zero** 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 (see 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 in a Client Side Accelerator configuration. See the "Displaying Domain Acceleration Records Statistics" section in Chapter 4, Configuring a Client-Side Accelerator.

Adding a DNS Name to a Content Rule

To specify a DNS name that maps to a content rule, use the **add dns** command. For zone-based DNS, use this command to configure domain names on a CSS acting as a **kal-ap** agent (for example, a CSS acting as a data center or a lower-level DNS server for an upper-level CSS configured with **kal-ap**, the CSS keepalive type). For details on **kal-ap**, see the "Configuring Domain Records" section.

Also, use the **add dns** command to specify domains on a CSS configured as a Content Routing Agent (CRA) VIP. For details on configuring a Content Routing Agent, see Chapter 3, Configuring a CSS as a Content Routing Agent.

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.

For example:

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

Removing a DNS Name 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 DNS Records with a Zero Weight

To provide backup sites in a DNS weighted roundrobin configuration when all domain records with weights from 1 to 10 are unavailable, configure DNS records with a weight of zero. When a DNS record has a weight of zero, a CSS does not consider that record for selection when using the weighted roundrobin algorithm unless all the other records, with weights from 1 to 10, are unavailable. This feature is intended especially for use in disaster recovery sites.

In a DNS peer mesh, it is possible to have multiple DNS records that have a weight of zero. If all the DNS records weighted from 1 to 10 are unavailable, then the CSS cycles through the zero-weighted DNS records using the roundrobin balance method.

You can configure the same DNS record with different balance methods on two sites. In this case, all decisions are based on the balance method that is configured on the CSS making the DNS decision. Even if a DNS record is not configured with weighted roundrobin, it still broadcasts a default weight of 1 to all its peers.

If you configure a default weight for all records in a zone, a CSS advertises that value for records that are part of the zone regardless of their balance method. You can override the default record weight for a zone by configuring a record with weighted roundrobin and assigning a weight value to the record using the **dns-record** command. For details on configuring a default record weight for a zone, see the "Configuring DNS Server Zones" section.

Note If DNS Sticky is enabled, a CSS could stick a user to a zero-weighted site. Even if other non-zero-weighted sites return to an active state, the CSS will not attempt to reroute the user to a non-zero site.

For details on the **dns-record** command including syntax, variables, and options, see the "Configuring Domain Records" section.

Configuring kal-ap-vip

The **kal-ap-vip** option of the **dns-record** command extends the functionality of **kal-ap** (the CSS keepalive that uses domain names configured on a single content rule) by providing load and status responses to queries for virtual IP (VIP) addresses configured on one or more content rules. This feature allows greater flexibility and accuracy of load and status reports for multiple content rules that are configured with the same VIP. This feature also eliminates the need for configuring domain names on a CSS that is responding to **kal-ap-vip** queries only and is not running a local DNS server.

Overview

In a manner similar to **kal-ap**, **kal-ap-vip** has two main components:

- Client
- Agent

A client is a CSS that requests load and status information for a VIP from an agent. You configure a client to generate queries using the **dns-record** command. For details, see the "Configuring a kal-ap-vip Client" section later in this section.

An agent is a CSS that responds to client queries with load and status reports for the requested VIPs. A **kal-ap-vip** agent can handle and respond to queries from local or remote CSSs (including itself) and other supported devices. No additional configuration is required for the agent.

To best service requests for a domain when a CSS makes GSLB decisions, a CSS may need to consider the keepalive status and load information of all content rules sharing the same VIP. Often, a **kal-ap-vip** configuration has at least two content rules to handle domain traffic: one for port 80 (TCP) and one for port 443 (SSL). The load reported by the agent is the average load of all the content rules that share the same VIP, unless a content rule is suspended.

In order for a **kal-ap-vip** agent to return a load value from 2 to 254 (indicating an Alive status) for a requested VIP, at least one service must be Up on each content rule sharing the requested VIP. For a requested VIP, if all services configured on one content rule are Down, or if one content rule is suspended, the agent reports a load of 255, indicating that the VIP is unavailable.

Configuration Requirements

Kal-ap-vip requires that you configure the following:

- Application Peering Protocol-User Datagram Protocol (APP-UDP) - Used to transmit **kal-ap-vip** datagrams. (For information on configuring APP-UDP, see the"Configuring APP-UDP and APP" section in Chapter 5, Configuring Network Proximity.) The datagrams can contain a mix of both kal-ap (by domain or tag) and **kal-ap-vip** requests.
- **dns-record** command with the **kal-ap-vip** option - Used to configure a **kal-ap-vip** client. See the "Configuring a kal-ap-vip Client" section later in this chapter.

Note You can configure **kal-ap-vip** and **kal-ap** on the same CSS. If you configure **kal-ap** on a CSS, you must also configure the **add dns** command with the appropriate domain names on the CSS acting as an agent. The agent responds with the load information for a VIP and/or a domain, as appropriate. For information on the **add dns** command, see the "Adding a DNS Name to a Content Rule" section.

Configuring a kal-ap-vip Client

To configure a **kal-ap-vip** client on a CSS to allow the CSS to query a **kal-ap-vip** agent for keepalive information on multiple content rules, use the **kal-ap-vip** option of the **dns-record** command.

The syntax for this global configuration command is:

dns-record a|ns *dns_name ip_address* {*ttl_value* {**single|multiple** {**kal-ap-vip** {*ip_address2*}}}}

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

- **a|ns** - Indicates a request for an address record (**a**) or a name server record (**ns**).
- *dns_name* - Domain name mapped to the address record or name server record. Enter the name as a lowercase unquoted text string with no spaces and a maximum 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). This is the VIP for which a CSS client sends a **kal-ap-vip** request to itself or another CSS agent for load information.
- *ttl_value* - 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- or NS-record. Specifying **multiple** returns two A- or NS-records.
- **kal-ap-vip** - Optional CSS keepalive message type keyword used by a CSS client to request load information for the VIP specified in the *ip_address* value from the CSS agent specified in the *ip_address2* value. Use this option to allow a CSS client to query a local or remote CSS agent for load information for a VIP configured on multiple content rules.
- *ip_address2* - IP address of the local or remote CSS agent 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.

For example:

```
(config)# dns-record a www.work.com 192.168.12.7 10 single
kal-ap-vip 172.16.25.3
```

For details on the other **dns-record** command options and variables, see the "Configuring Domain Records" section.

Configuring Content Rule-Based DNS on a CSS

Content rule-based DNS uses domain names configured on content rules to resolve DNS requests to IP addresses. Such a configuration is sometimes referred to as *static proximity.* Each CSS:

- Acts as an authoritative DNS server for the domain it represents
- Shares owner information, content rules, and load information with other CSS peers using APP

This section contains the following topics:

Note The recommended method for configuring DNS in a global server load balancing environment on a CSS is zone-based DNS. For details on configuring zone-based DNS server functionality on a CSS, see the "Configuring Zone-Based DNS on a CSS" section.

Content Rule-Based DNS Quick Start

Table 1-4 provides a quick overview of the steps required to configure content rule-based DNS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 1-4.

Table 1-4 Content Rule-Based DNS Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Enable APP to allow the local CSS to communicate with remote CSSs. See the "Configuring the Application Peering Protocol" section. `(config)# app`

Table 1-4 Content Rule-Based DNS Configuration Quick Start (continued)

Task and Command Example
3. Configure an APP session with any remote CSS that is participating in the peer mesh with the local CSS. The IP address you enter is a local interface address on the remote CSS. See the "Configuring the Application Peering Protocol" section. Be sure to configure APP sessions on the remote CSSs also. `(config)# app session 172.16.2.5`
4. Configure the CSS to act as a DNS server. See the "Enabling a DNS Server" section. `(config)# dns-server`
5. (Optional) Configure the **dns-peer** command and its options to set: • Interval between load reports • Receive-slots • Send slots See the "Configuring CSS DNS Peering" section. `(config)# dns-peer interval 60`
6. (Optional) Set the DNS exchange policy for the owner. See the "Configuring the DNS Exchange Policy for an Owner" section. `(config-owner[arrowpoint])# dns both`
7. Use the **add dns** command to configure: • Domain that maps to a content rule • TTL for the domain name See the "Adding a DNS Name to a Content Rule" section. `(config-owner-content[arrowpoint-rule1])# add dns www.arrowpoint.com 36`
8. (Optional) Verify the configuration using the **show** commands described in the sections that follow this table.

The following running-config example shows the results of entering the commands described in Table 1-4.

```
!************************** GLOBAL ***************************
  dns-server
  dns-peer interval 60

  app
  app session 172.16.2.5

!************************** OWNER ****************************
owner arrowpoint
  dns both

  content rule1
    add dns www.arrowpoint.com 36
```

Configuring the DNS Exchange Policy for an Owner

To set the DNS exchange policy for an owner, use the **dns** command. This command specifies how a CSS exchanges owner information and content rules with DNS peers. This functionality is disabled by default.

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 pushes all its content rules to the CSS peer.
- **dns both** - Advertises the owner and pushes all its content rules to the CSS peer, and accepts all this owner's content rules proposed by the CSS peer.

For example, enter:

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

To reset the default CSS behavior of no exchange policy, enter:

```
(config-owner[arrowpoint])# no dns
```

Configuring CSS DNS Peering

To configure DNS peer functionality on a CSS, use the **dns-peer** command and its options. Peering specifies how CSSs share domain names and load information. This functionality is automatically enabled when you configure APP. See the "Configuring the Application Peering Protocol" section.

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

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

Configuring the DNS Peer Interval

To set the time between sending 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 a 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 a 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 more than 128 DNS names to each 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
```

Configuring DNS Peer Load Variance

To set the deterministic difference in peer load numbers that a CSS considers to be similar for the leastloaded algorithm in a DNS load-balancing decision., use the **dns-peer load variance** command.

The syntax for this global configuration mode command is:

dns-peer load variance *number*

For the *number* variable, enter an integer from 0 to 254. The default is 50. A value of zero disables the two-stage comparison used to determine the least-loaded site.

For example:

```
(config)# dns-peer load variance 200
```

To reset the DNS peer load variance to its default value of 50, enter:

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

Configuring a DNS Server

As a DNS server, a CSS resolves domain names to IP addresses when it receives DNS requests from clients. To configure a CSS as a DNS server, use the **dns-server** command and its options. The options for this global configuration mode command are:

- **dns-server** - Enables the DNS server functionality on a CSS.
- **dns-server bufferCount** - Modifies the DNS response buffer count.

- **dns-server respTasks** - Modifies the DNS responder task count.
- **dns-server forwarder** - Enables a DNS forwarder (a CSS or a fully-functional BIND server), which resolves DNS requests that a CSS cannot resolve.

Note You need to configure APP before you configure the DNS server as shown in Table 1-4.

Enabling a DNS Server

To enable the DNS server functionality on a CSS, use the **dns-server** command.

Note The **dns-server** command is part of the CSS Enhanced feature set and requires a separate license key.

For example:

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

To disable DNS server functionality on a CSS, enter:

```
(config)# no 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 and the oldest requests are dropped.

For example:

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

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

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

Configuring a DNS Forwarder

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 forwarder, can be any DNS server or a CSS configured as a DNS server. The CSS sends to the forwarder DNS requests that:

- Cannot be resolved by the CSS
- Contain an unsupported request or record type

Note For Client Side Accelerator (CSA) configurations, the forwarder must be a full-featured DNS server. For details about CSA, see 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.

To configure a DNS forwarder on a CSS, use the **dns-server forwarder** command.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 requests that it cannot resolve 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
```

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
```

Adding a DNS Name 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 a DNS Name 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 ?
```

Displaying CSS DNS Information

Use the **show** commands in the following sections to display DNS information for a CSS configured as a DNS server. This section contains the following topics:

- Displaying DNS Server Information
- Displaying DNS Server Zones
- Displaying DNS Record Information
- Displaying DNS Peer Information
- Displaying Domain Summary Information

Displaying DNS Server Information

To display DNS server configuration and database information, use the **show dns-server** commands for both zone-based and rule-based DNS configurations. These commands provide the following options and information:

- **show dns-server** - Displays DNS server configuration information
- **show dns-server dbase** - Displays DNS database information
- **show dns-server stats** - Displays DNS database statistics
- **show dns-server forwarder** - Displays DNS server forwarder statistics

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

Table 1-5 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.
Name Server Buffers	
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 currently available.

Table 1-5 Field Descriptions for the show dns-server Command (continued)

Field	Description
Minimum Free Count	The smallest number of buffers that were available at any one time.
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.
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 because the domain was not configured or no resources were online.
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.
Internal Resolver	
Requests Sent	The number of queries sent to another name server for resolution.
Responses Accepted	The number of replies received from another name server.
Proximity Lookups	
Requests Sent	The number of proximity lookups sent to the PDB.
Responses Accepted	The number of proximity responses received from the PDB.

Note Proximity lookup information is displayed only when you configure a PDB IP address. For information on configuring a PDB, see the "Configuring a Proximity Database" section in Chapter 5, Configuring Network Proximity.

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 1-6 describes the fields in the **show dns-server dbase** output.

Table 1-6 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. For details, refer to the **host** command in the *Cisco Content Services Switch Administration Guide*.

Displaying DNS Server Domain Statistics

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

show dns-server stats

Table 1-7 describes the fields in the **show dns-server stats** output.

Table 1-7 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 Forwarder Statistics

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

show dns-server forwarder

Table 1-8 describes the fields in the **show dns-server forwarder** output.

Table 1-8 Field Descriptions for the show dns-server forwarder Command

Field	Description
DNS Server Forwarder Primary	The state of the primary forwarder. The states are: • Not Configured • Up • Down
DNS Server Forwarder Secondary	The state of the secondary forwarder. The states are: • Not Configured • Up • Down
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.
Totals:	
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* - Displays the zone index of a peer. If you omit this variable, this command displays the states of all proximity zones.
- **local** - Displays 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** - Displays 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 1-9 describes the fields in the **show zone** output.

Table 1-9 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 **dns-server zone** command.

Table 1-9 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: • **INIT** - Initializing. Waiting for local configuration to complete. • **SREQ** - A connection request message has been sent to the peer. • **RACK** - An acknowledgment message has been received from the peer. • **SACK** - An acknowledgment request has been sent to the peer. • **OPEN** - Negotiations with the peer have completed successfully and the connection is open. • **CLOSED** - Negotiations with the peer have failed and the connection is closed.
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.

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 in a zone-based DNS configuration.

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. For information about accelerated domain records, see Chapter 4, Configuring a Client-Side Accelerator.

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

Table 1-10 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: • **A** - Address record • **NS** - Name-server record • **Accel** - An accelerated domain associated with a Client Side Accelerator (CSA)
IP Address	Configured IP address for the zone.

Table 1-10 Field Descriptions for the show dns-record statistics Command (continued)

Field (continued)	Description
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 1-11 describes the fields in the **show dns-record keepalive** output.

Table 1-11 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, ICMP, kal-ap, or none.
IP	Destination IP address of the keepalive message.
State	State of the record, either UP or DOWN.
Transitions	Number of state transitions.

Table 1-11 Field Descriptions for the show dns-record keepalive Command (continued)

Field	Description
Load	Load for the record, which applies only to a kal-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 a kal-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 1-12 describes the fields in the **show dns-record weight** output.

Table 1-12 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 within the DNS server zone.

Table 1-12 Field Descriptions for the show dns-record weight Command (continued)

Field	Description
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.

Displaying DNS Peer Information

To display the DNS peering configuration, use the **show dns-peer** command in a rule-based DNS configuration.

For example:

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

Table 1-13 describes the fields in the **show dns-peer** output.

Table 1-13 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.

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 hot lists.
- **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 1-14 describes the fields in the **show domain** output.

Table 1-14 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 APP (CCC) link.
OWN State	The state of the owner policy negotiation FSM that determines the owners about whom the peers share domain name and rule information.

Table 1-14 Field Descriptions for the show domain Command (continued)

Field	Description
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 sends load reports to the peer.
ReceiveSlots	The number of individual domain name rules on which the CSS receives 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 sends load reports to the peer.
Receiving Load Reports for	The list of domain names for which the CSS receives 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.