Configuring Source Groups, ACLs, EQLs, URQLs, NQLs, and DQLs

This chapter describes how to configure source groups, Access Control Lists (ACLS), Extension Qualifier Lists (EQLs), Uniform Resource Locator Qualifier Lists (URQLs), Network Qualifier Lists (NQLs), and Domain Qualifier Lists (DQLs). Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following sections:

- Configuring Source Groups
- Configuring an Access Control List
- Configuring Extension Qualifier Lists
- Configuring Uniform Resource Locator Qualifier Lists
- Configuring Network Qualifier Lists
- Configuring Domain Qualifier Lists

Configuring Source Groups

Group configuration mode allows you to configure a maximum of 255 source groups on a CSS. A source group is a collection of local servers that initiate flows from within the local web farm. The CSS enables you to treat a source group as a virtual server with its own source IP address.

For example, if you configure several streaming audio transmitters as a group, the CSS will process flows from the group members and give them all the same source IP address.

This section covers:

- Source Group Configuration Quick Start
- Creating a Source Group
- Configuring a Source Group for FTP Connections
- Configuring Source Groups to Allow Servers to Internet-Resolve Domain Names
- Showing Source Groups
- Clearing Source Group Counters

Source Group Configuration Quick Start

Use the procedure in Table 5-1 to configure a source group for TCP/UDP traffic. To configure a source group for FTP traffic, see the next section. Note that each source group requires a content rule that contains the same services and VIP as the source group.

Table 5-1 Source Group Configuration Quick Start

Task and Command Example

1. Create the source group. Source group names can be a maximum of 16 characters. The following example creates a source group *ftpgroup*.
`(config)# group ftpgroup`
The CLI transitions into config-group mode where you can activate the source group and configure attributes for it.
`(config-group[ftpgroup])#`

2. Configure the source group VIP address to which all service IP addresses will be translated. You can assign the same VIP address to multiple source groups, but only one of the source groups can be active at a time. For example, enter: `(config-group[ftpgroup])# vip address 172.16.36.58`

3. Add previously defined services to the source group. For example, enter:
`(config-group[ftpgroup])# add service server1`
`(config-group[ftpgroup])# add service server2`

4. Activate the source group. Because a VIP address can belong only to one active source group at a time, the CSS will not allow you to activate a second source group that contains the same VIP address as the one in the active source group.
`(config-group[ftpgroup])# active`
To remove service *server1* from the source group, enter:
`(config-group[ftpgroup])# remove service server1`

Table 5-1 Source Group Configuration Quick Start (continued)

Task and Command Example

5. Create a content rule, add the same services and VIP that are configured in the source group, and activate the content rule. The content rule enables the CSS to match requests for the content rule VIP. When either *server1* or *server2* replies to the request, the CSS NATs the server IP addresses to the source group VIP.
 	
For example, enter:
 	
`(config-owner[arrowpoint.com])# content ftpsource1`
 	
`(config-owner-content[arrowpoint.com-ftpsource1])# add service server1`
 	
`(config-owner-content[arrowpoint.com-ftpsource1])# add service server2`
 	
`(config-owner-content[arrowpoint.com-ftpsource1])# vip address 172.16.36.58`
 	
`(config-owner-content[arrowpoint.com-ftpsource1])# activate`

Creating a Source Group

To access group configuration mode, use the **group** command from any mode except ACL and boot configuration modes. The syntax for this command is:

group *groupname*

Enter an existing or a new source group name from 1 to 31 characters.

For example, enter:

```
(config)# group ftpgroup
(config-group[ftpgroup])#
```

To view a list of existing source groups, enter:

```
(config)# group ?
```

Note You can also use the **group** command from within group mode to access or create another source group.

To remove a source group, enter:

```
(config)# no group ftpgroup
```

Note To make certain modifications to an active source group, you must first suspend the source group using the **suspend** command. Such modifications include: changing the IP address to 0 or using the **no ip address command**, adding or removing a service or destination service, or using the **portmap** command.

Source Group Commands

Use the following commands in group mode:

- **active** - Activates a source group.
- **add destination service** *service_name* - Adds a destination service to a source group. You can configure a maximum of 64 destination services per source group. You cannot use a service with the same name in other source groups or the source service list within the same source group. You can use services with duplicate addresses among destination services because the actual service is chosen through content rule selection. The destination service must be active and must be added to a content rule in order for it to perform destination source address NATing for the source group (refer to Chapter 3, Configuring Content Rules).

  Note Adding a destination service to a source group does not allow the destination service flows to be NATed by the source group, when the service initiates the flows. This is because the destination service applies group membership based on rule and service match. To ensure that service initiated connections are NATed, you must also configure ACL match criteria or additional service names with duplicate addresses, then add those services to a source group. The source group used could be the current source group with the destination service or any other configured source group.

- **add service** - Adds a service to a source group. You can configure a maximum of 64 services per source group. A service may belong to only one group at a time. When the source group is active and the same service is hit through a content rule, ACL preferred service, or sorry service, the source group is used to NAT (Network Address Translation) the source address. The service must be active in order for it to perform source address NATing for the source group (refer to Chapter 1, Configuring Services).

  Be aware that you cannot use a service with:

  - The same name in other source groups or the destination service list within the same source group
  - The same address as a source service on another source group

- **vip address** - Specifies the source Virtual IP address (VIP) for the group. The CSS substitutes this IP address for the source address in flows originating from one of the group's sources. You can assign the same VIP address to multiple source groups, but only one of the source groups can be active at a time.
- **remove service** - Removes a previously configured service from a source group.
- **portmap** - Defines the source port translation of flows from the services configured in a source group. By default, portmapping is enabled for source groups on source ports greater than 1023. The CSS translates such source ports to a range starting at 8192. Use the following portmap options to change the default portmapping behavior of the CSS. The syntax and options for this group mode command are:

  - **portmap base-port** *base_number* - Defines the base port (starting port number) for the CSS. Enter a base number from 2016 to 63456. The default is 2016.

    To reset the starting port number to its default value of 2016, use the **no portmap base-port** command.

  - **portmap number-of-ports** *number* - Defines the number of ports in the portmap range for each Switch Processor (SP) in an 11500 series CSS or a Switch Fabric Processor (SFP) in an 11000 series CSS. Enter a number from 2048 to 63488. The default is 63488.

    To reset the number of ports to the default value, use the **no portmap number-of-ports** command.

– **portmap disable** - Instructs the CSS to perform Network Address Translation (NAT) only on the source IP addresses and not on the source ports of UDP traffic hitting a particular source group. Use this option for Wireless Application Protocol (WAP) or other applications where you need to preserve the registered UDP port number for return traffic.

Note This command does not affect TCP flows.

The CSS maintains but ignores any **base-port** or **number-of ports** (see the previous options) values configured in the source group. If you later reenable portmapping for that source group, any configured **base-port** or **number-of ports** values will take effect. The default behavior for a configured source group is to NAT both the source IP address and the source port for port numbers greater than 1023.

To restore the default CSS behavior of NATing source IP addresses *and* source ports for a configured source group, use the **portmap enable** command.

- **suspend** - Suspends a source group. The group and its attributes remain the same but no longer have an effect on flow creation.

Configuring a Source Group for FTP Connections

To use source groups to support FTP sessions to a VIP that is load balanced across multiple services, configure a content rule for the VIP and then a source group.

Note When you use an FTP content rule with a configured VIP address range, be sure to configure the corresponding source group with the same VIP address range (refer to Chapter 3, Configuring Content Rules).

To configure FTP sessions to a VIP:

1. Configure a content rule as required using the VIP that will be load balanced across multiple servers. The following example shows the portion of a running-config for content rule *ftp_rule*. Ensure that you use the **application ftp-control** command to define the application type.

```
content ftp_rule
    vip address 192.168.3.6
    protocol tcp
    port 21
    application ftp-control
    add service serv1
    add service serv2
    add service serv3
    active
```

2. Configure a source group defining the same VIP and services as configured in the content rule.

Note If you are load-balancing passive FTP servers, you must configure services directly in the associated source groups as shown in the following example. Active FTP does not require that you configure services in source groups.

The following running-config example shows source group *ftp_group*.

```
group ftp_group
    vip address 192.168.3.6
    add service serv1
    add service serv2
    add service serv3
    active
```

Configuring Source Groups to Allow Servers to Internet-Resolve Domain Names

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

To configure a server to resolve domain names:

1. If you have not already done so, configure the server.

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

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

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

   The following example creates content rule *dns1* with a public VIP 192.200.200.200 and adds server *Server1*.

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

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

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

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

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

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

Showing Source Groups

To display source group configuration information, use the **show group** commands in SuperUser, User, Global Configuration, and Group modes. The options are:

- **show group** - Display all source group configurations
- **show group** *group_name* - Display the source group configuration specified by group_name
- **show group** *group_name* **portmap** - Display the starting port number and number of ports configured on each SP in a 11500 series CSS (or SFP in a 11000 series CSS)

For example, enter:

```
(config)# show group
```

Table 5-2 describes the fields in the **show group** output.

Table 5-2 Field Descriptions for the show group Command

Field	Description
Group	The name of the group, whether the group is activated (Active) or suspended (Suspend), and the source IP address for the group.
Session Redundancy	Indicates whether Adaptive Session Redundancy (ASR) is enabled or disabled for the source group. For details on ASR, refer to the *Cisco Content Services Switch Advanced Configuration Guide*.
Redundancy Global Index	The unique global index value for Adaptive Session Redundancy assigned to the source group using the **redundant-index** command in group configuration mode.
Associated ACLs	Any ACLs associated with the group.
Source/Destination Services	The source or destination services of the source group.
Name	The name of the service.
Hits	The number of content hits on the service. This field is incremented for traffic from a group server going out from the source group. Traffic coming into the group does not increment the counter.
State	The state of the service. The possible states are Alive, Dying, or Dead.
DNS Load	The DNS load for the service. A load of 255 indicates that the service is down. An eligible load range is from 2 to 254.

Table 5-2 Field Descriptions for the show group Command (continued)

Field	Description
Trans	The number of times that the state of the service has transitioned.
Keepalive	The keepalive type of the service. The possible types are FTP, HTTP, ICMP, NAMED, SCRIPT, or TCP.
Conn	The number of connections currently on the service.
Flow Timeout Multiplier	Number of seconds that a flow remains idle before the CSS reclaims the flow resources, as configured with the **flow-timeout-multiplier** command. For details on the **flow-timeout-multiplier** command, refer to the *Cisco Content Services Switch Administration Guide*.
Group Cumulative Counters	The counters for the group.
Hits/Frames/Bytes	The number of group hits, frames, and bytes. This field is incremented for traffic from a group server going out from the source group. Traffic coming into the group does not increment the counter.
Connection Total/Current	The total number of connections and the current number of connections for the group.
FTP Control Total/Current	The total number of FTP control channels that were mapped and monitored by the CSS, and the current number of those connections that are mapped.
SP (or SFP) Port Map Info	The port map information for each SP in the 11500 series CSS (or SFP in the 11000 series CSS). Includes the status of the **portmap** command (Enabled or Disabled).
SP (or SFP)	The slot and port number of the SP in the 11500 series CSS (or SFP in the 11000 series CSS).
Base Port	The starting SP port number in the 11500 series CSS (or SFP port number in the 11000 series CSS) in the chassis.

Table 5-2 Field Descriptions for the show group Command (continued)

Field	Description
Configured Base Port	The configured starting port number.
Configured Ports SP (or SFP)	The configured number of ports allowed on each SP in the 11500 series CSS (or SFP in the 11000 series CSS).
Current Mapped Ports	The current number of mapped ports.
Last Mapped Port	The most recently mapped port number for each SP in the 11500 series CSS (or SFP in the 11000 series CSS).
High Water Mark	The highest number of ports that this source group has had concurrently mapped since the last group was activated.
No Portmap Errors	The number of times no port could be allocated by the portmapper.

Clearing Source Group Counters

To set the statistics displayed by the **show group** command to zero, use the **zero all** command. The reset counter statistics appear as zero in the **show group** display.

For example, enter:

```
(config-group[ftpgroup])# zero all
```

Configuring an Access Control List

The following sections describe how to configure an Access Control List (ACL):

- Access Control List Overview
- Creating an ACL
- Creating an ACL
- Deleting an ACL
- Configuring Clauses
- Deleting a Clause
- Logging ACL Activity
- Applying an ACL to a Circuit or DNS Queries
- Removing an ACL from a Circuit or DNS Queries
- Globally Enabling ACLs
- Showing ACLs
- ACL Example

Access Control List Overview

The CSS provides traffic filtering capabilities with Access Control Lists (ACLs). ACLs filter network traffic by controlling whether packets are forwarded or blocked at the CSS interfaces. You can configure ACLs for routed network protocols, filtering the protocol packets as the packets pass through the CSS.

An ACL consists of clauses that you define. The CSS uses these clauses to determine how to handle each packet it processes. When the CSS examines each packet, it either forwards or blocks the packet based on whether or not the packet matches a clause in the ACL.

Note ACLs are not supported on the CSS Ethernet Management port.

The total number of ACL hits for each packet received by the CSS can vary depending on the type of flow and whether an ACL match occurred. The CSS performs an ACL check for every packet received until the flow is completely set up.

- For Content Hits, a flow can be defined as a stream of UDP and TCP packets between a client and a server. The CSS must receive a number of packets from the client and the server before it can completely set up the flow. All of these packets, received before the flow is completely set up, are subject to ACL checks and can cause increments to the ACL Content Hits counter.
- For Router Hits, all non-TCP or UDP packets subjected to ACL checks cause increments to the ACL Router Hits counter. All UDP and TCP traffic terminating on the CSS (for example, a Telnet or FTP session) cause increments to the ACL Router Hits counter.

ACLs provide a basic level of security for accessing your network. If you do not configure ACLs on the CSS, all packets passing through the CSS could be allowed onto the entire network. For example, you may want to permit all email traffic, but block Telnet traffic. You can also use ACLs to allow one client to access a part of the network and prevent another client from accessing the same area.

Caution ACLs function as a firewall security feature. When you enable ACLs, all traffic not configured in an ACL permit clause *will be denied*. It is extremely important that you first configure an ACL to permit traffic *before you enable ACLs*. If you do not permit any traffic, you will lose network connectivity. Note that the console port is not affected.

Cisco recommends that you configure either a permit all or a deny all clause depending on your ACL configuration. For example, you could first configure a permit all clause and then configure deny clauses for only the traffic you wish to deny. Or, use the default deny all clause and configure permit clauses only for the traffic you wish to permit.

ACL Configuration Quick Start

Use the procedure in Table 5-3 to configure an ACL. Each step includes the CLI command required to complete the task. For a complete description of each feature, see the sections following this procedure.

Table 5-3 ACL Configuration Quick Start

Task and Command Example

1. Create an ACL and access ACL mode. Define the ACL index number from 1 to 99.
`(config)# acl 7`
`(config-acl[7])#`

2. To control traffic on a circuit, configure clauses in the ACL. Enter a clause number from 1 to 254 and define the clause parameters. The syntax for defining a clause is: **clause** *number* **permit**|**deny**|**bypass** *protocol* [*source_info* {*source_port*}] **dest** [*dest_info* {*dest_port*}] {**log**} {**prefer** *servicename*} {**sourcegroup** *name*} For example, enter: `(config-acl[7])# clause 1 deny udp any eq 3 dest any eq 3 log prefer serv7` If you are load-balancing passive FTP servers and you want to use an ACL to apply a source group, you must configure services directly in the source group. For details on using source groups to support FTP sessions, see "Configuring a Source Group for FTP Connections" earlier in this chapter.
3. Apply the ACL to a specific circuit or add the ACL to DNS queries. For example, to apply acl 7 to circuit VLAN1, enter: `(config-acl[7])# apply circuit-(VLAN1)`
4. Enable all ACLS on the CSS. Enter the global **acl enable** command for all ACLs to take effect. You can enable ACL mode even if no ACLs are configured. When you enable ACLs, all traffic not specifically permitted in an ACL permit clause is **denied** by default. For example, enter: `(config)# acl enable`

Caution When you enter the **acl enable** command, all traffic is denied except for traffic specified in an ACL permit clause.

Creating an ACL

To create an ACL and access ACL mode, use the **acl** *index number* command. The index number defines the ACL and can range from 1 to 99. To display a list of existing ACLs, use the **acl ?** command.

```
(config)# acl 7
```

When you access this mode, the prompt changes to the ACL mode of the index number you created. For example, enter:

```
(config-acl[7])#
```

Deleting an ACL

To delete an ACL, use the **no acl** command followed by the index number you wish to delete. For example, enter:

```
(config)# no acl 2
```

Configuring Clauses

To control traffic on a circuit, the CSS enables you to enter clauses in a specific ACL. When implementing an ACL, the number assigned to each clause is very important. The CSS looks at the ACL starting from clause 1 and sequentially progresses through the rest of the clauses. Assign the lowest clause numbers to clauses with the most specific matches. Then, assign higher clause numbers to clauses with less specific matches.

You do not need to enter the clauses sequentially. The CSS automatically inserts the clause in the appropriate order in the ACL. For example, if you enter clauses 10 and 24, and then clause 15, the CSS inserts the clauses in the correct sequence.

Clause *number* is the number you want to assign to the clause. Enter a number from 1 to 254. To create a clause to permit, deny, or bypass traffic on a circuit, use the **clause** command.

Note Ensure that ACLs associated with a source group specified in the **clause** command are globally enabled for the ACL to properly map to the source group (see "Globally Enabling ACLs" later in this chapter).

The syntax for the **clause** command is:

- **clause** *number* **bypass** - Creates a clause in the ACL to *permit* traffic on a circuit and *bypass* (do not apply) content rules that apply to the traffic. The syntax for **clause bypass** is:

  clause *number* **bypass** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

  Note The **bypass** option bypasses traffic only on a content rule, thus does not cause NATing to occur. Do not use the **bypass** option in an ACL clause with a source group. Since this option does not bypass traffic that does not match a rule, it does not effect NATing on a source group in an ACL clause.

- **clause** *number* **deny** - Creates a clause in the ACL to *deny* traffic on a circuit. The syntax for **clause deny** is:

  clause *number* **deny** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

- **clause** *number* **permit** - Creates a clause in the ACL to *permit* traffic on a circuit. When you configure an ACL permit clause, all traffic not specified in a permit clause is denied by default. The syntax for **clause permit** is:

  clause *number* **permit** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

Note If you specify both a source group and a preferred service in a clause, you must specify the source group before you specify the preferred service within the clause.

Table 5-4 provides variables and options for the **clause** command. Bolded syntax defines keywords that you enter on the command line. Italics define variables where you enter a value such as an IP address or host name.

Note ACLs are not supported on the CSS Ethernet Management port.

Note When a destination in an ACL clause is a Layer 5 content rule, the CSS does not spoof the connection. Therefore, the ACL clause does not function as would be expected. As a workaround, you may configure an additional clause to permit the TCP IP addresses and ports. Be aware that content will be matched on both clauses. For example,
clause 14 permit any any destination content Layer5/L5 eq 80 (original clause)
clause 15 permit tcp any destination 200.200.200.200 eq 80 (This is an additional clause to handle the SYN, where the destination IP address is the IP address configured in the Layer 5 content rule. Note that this clause number must be greater than the destination content clause number.)

Table 5-4 Clause Command Options

Variables and Options	Parameters
number	The number you want to assign to the clause. Enter a number from 1 to 254.
action	The action to apply to the clause. Enter one of the following: **bypass**, **deny**, **permit**.
protocol	The protocol for the traffic type. Enter one of the following: **any**, **icmp**, **igp**, **igmp**, **ospf**, **tcp**, **udp**.

Table 5-4 Clause Command Options (continued)

Variables and Options	Parameters
source_info	The source of the traffic. Enter one of the following: • *ip_address* (optionally include *subnet mask* in IP address format only) for the source IP address and optional mask IP address. • *hostname* for the source host name. Enter a host name in mnemonic host-name format. Configure the CSS DNS client first to enable the CSS to translate the host name. • **any** for any combination of source IP address and host name information. • **nql** *nql_name* for an existing Network Qualifier List (NQL) consisting of a list of IP addresses.
source_port	The source port for the traffic. If you do not designate a source port, this clause allows traffic from any port number. Enter one of the following: • **eq** *port* is equal to the port number. • **lt** *port* is less than the port number. • **gt** *port is* greater than the port number. • **neq** *port* is not equal to the port number. • **range** *low high* for a range of port numbers, inclusive. Enter numbers from a range of 1 to 65535. Separate the *low* and *high* number with a space.

Table 5-4 Clause Command Options (continued)

Variables and Options	Parameters
destination_info	The destination information for the traffic. Enter one of the following: • **destination any** for any combination of destination information. • **destination content** *owner_name*|*rule_name* for an owner content rule. Separate the owner and rule name with a \ character. • **destination** *ip_address* (for the destination IP address and optional subnet mask IP address. Include *subnet mask* as IP address only, no CIDR. • **destination** *hostname* for the destination host name. To use a *hostname*, configure the CSS DNS client first to enable the CSS to translate the host name. • **nql** *nql_name* for an existing NQL consisting of host IP addresses. Enter the name of the NQL.

Table 5-4 Clause Command Options (continued)

Variables and Options	Parameters
destination_port	The destination port. Enter one of the following. You may use a port number or port name with the options. • **eq** *port* is equal to the port number. • **lt** *port* is less than the port number. • **gt** *port* is greater than the port number. • **neq** *port* is not equal to the port number. • **range** *low high* for a range of port numbers, inclusive. Enter numbers from a range of 1 to 65535. Separate the *low* and *high* number with a space. • *port names*: **https** = Port 443 Https, **ldap** = Port 389 Ldap, **bgp** = Port 179 Bgp, **ntp** = Port 123 Ntp, **nntp** = Port 119 Nntp, **pop** = Port 110 Pop, **http** = Port 80 Http, **gopher** = Port 70 Gopher, **domain** = Port 53 Domain, **smtp** = Port 25 Smtp, **telnet** = Port 23 Telnet, **ftp** = Port 21 Ftp, **ftp-data** = Port 20 Ftp-data, **none** = None If you do not define a destination port, this clause allows traffic to any port.

Table 5-4 Clause Command Options (continued)

Variables and Options	Parameters
sourcegroup *name*	Define a source group based on matching this ACL clause. Enter the group name. To see a list of source groups, enter: `show group ?`
prefer *service_name*	Define a preferred service based on matching the ACL clause. Enter the service name. To define more than one preferred service, separate each service with a comma (,). You can define a maximum of two services. You cannot configure services learned through an Application Peering Protocol (APP) session as preferred services. A remote service learned through APP is of the form ap-redirect@207.140.138.118 and can been seen on the **show service summary** screen. When configuring an ACL clause, you cannot use this service as a preferred service. If you save this clause in the startup-config and reboot the CSS, a startup error occurs because this service has not been learned through APP at this point. For example, enter: `clause 10 permit any any destination any prefer ap redirect@207.140.138.118`

Deleting a Clause

To delete a clause, use the **no clause** command. For example, enter:

```
(config-acl[7]) no clause 6
```

Logging ACL Activity

When you configure the CSS to log ACL activity, it logs the event of the packet matching the clause and ACL. The CSS sends log information to the location you specified in the **logging** command. For information on the **logging** command, refer to the *Cisco Content Services Switch Administration Guide*.

Note Before you configure logging for a specific ACL clause, ensure that global ACL logging is enabled. To globally enable ACL logging, use the **(config)# logging subsystem acl level debug-7** command.

Because the CSS does not save the clause log enable command in the running-config, you must reenable logging if the CSS reboots.

To configure logging for an ACL clause:

1. Enter the ACL mode for which you want to enable logging.

```
(config)# acl 7
(config-acl[7])#
```

2. Remove the ACL from the circuit. You must remove an ACL from a circuit before making any clause changes.

```
(config-acl[7]) remove circuit-(VLAN1)
```

3. Enable logging for the existing clause.

```
(config-acl[7])# clause 1 log enable
```

4. Reapply the ACL to the circuit.

```
(config-acl[7])# apply circuit-(VLAN1)
```

To disable ACL logging for a specific clause, enter:

```
(config-acl[7])# clause 1 log disable
```

To globally disable logging for all ACL clauses, enter:

```
(config)# no logging subsystem acl
```

Applying an ACL to a Circuit or DNS Queries

Once you configure the ACL, use the **apply** command to assign an ACL to all circuits, an individual circuit, or to DNS queries.

Note You cannot apply an empty ACL to a circuit. If you attempt to do so, the error message `Cannot apply ACL for it has no clauses` appears.

To add a new clause to an existing and applied ACL, reapply the ACL to the circuit with the **apply circuit** command.

To apply any changes to an existing clause on an existing and applied ACL, you must remove the ACL from the circuit with the **(config-acl) remove** command, and then reapply the ACL to the circuit.

To remove a clause currently in use, you must remove its applied ACL from the circuit, delete the clause, and then reapply the ACL to the circuit.

The syntax and options for this ACL mode command are:

- **apply all** - Applies the ACL to all existing circuits
- **apply circuit** - (*circuit_name*) - Applies the ACL to an individual circuit
- **apply dns** - Adds the ACL to DNS queries

Note If you configure a CSS with the **dns-server** command, and the CSS receives a DNS query for a domain name that you configured on the CSS using the **host** command, the DNS query *will not* match on an ACL that is configured with the **apply dns** command.

However, if you configure a domain name on a content rule on a CSS using the **add dns** *domain_ name* command, a DNS query for that domain name *will* match on an ACL that is configured with the **apply dns** command.

For example, to apply acl 7 to circuit VLAN1:

```
(config-acl[7])# apply circuit-(VLAN1)
```

To display a list of circuits, use the **apply ?** command.

Note You must enter the global **acl enable** command for ACLs to take effect. For information on the **acl enable** command, see "Globally Enabling ACLs" later in this chapter.

Removing an ACL from a Circuit or DNS Queries

Use the **remove** command to remove an ACL from all circuits, an individual circuit, or from DNS queries.

Note To remove a clause currently in use, you must remove its applied ACL from the circuit, delete the clause, and then reapply the ACL to the circuit.

The syntax and options for this ACL mode command are:

- **remove all** - Removes the ACL from an individual circuit. To display a list of circuits that you can remove, use the **remove ?** command.
- **remove circuit** (*circuit_name*) - Removes the ACL from a circuit. To display a list of circuits that you can remove, use the **remove ?** command.
- **remove dns** - Removes the ACL from DNS queries.

For example, enter:

```
(config-acl[7])# remove circuit-(VLAN1)
(config-acl[7])# remove dns
```

Globally Enabling ACLs

Global ACL commands allow you to enable or disable all ACLs simultaneously. Global commands are advantageous when managing your network.

Caution When you enter the **acl enable** command, all traffic is denied except for traffic specified in an ACL permit clause.

To globally enable all ACLs, enter:

```
(config)# acl enable
```

To globally disable all ACLs on the CSS, enter:

```
(config)# acl disable
```

Showing ACLs

Use the **show acl** commands to display access control lists and clauses. The **show acl** commands are available in all modes.

When you show an ACL clause that is applied to a circuit, the display includes:

- **Content Hits** - A flow can be defined as a stream of UDP and TCP packets between a client and a server. The CSS must receive a number of packets from the client and the server before it can completely setup the flow. All of these packets, received before the flow is completely setup, are subject to ACL checks and can cause increments to the ACL Content Hits counter.
- **Router Hits** - All non-UDP and -TCP packets subjected to ACL checks cause increments to the ACL Router Hits counter. All UDP and TCP traffic terminating on the CSS (for example, a Telnet or FTP session) cause increments to the ACL Router Hits counter.

When you show an ACL clause that is applied to DNS queries, the display includes a DNS hit counter, which counts DNS lookups.

The syntax is:

- **show acl** - Displays all ACLs and their clauses.
- **show acl** *index* - Displays the clauses for the specified ACL index number (valid numbers are 1 to 99).
- **show acl config** - Shows the ACL global configuration. This display also shows you which ACLs are applied to which circuits.

For example, enter:

```
(config)# show acl 2
```

Table 5-5 describes the fields in the **show acl** output.

Note The total number of ACL hits for each packet received by the CSS can vary depending on the type of flow and whether an ACL match occurred. The CSS performs an ACL check for every packet received until the ACL flow is completely setup. Once the ACL flow is setup, remaining packets received by the CSS that are associated with the flow are not subject to an ACL match and the ACL hit counters do not increment.

Table 5-5 Field Descriptions for the show acl Command

Field	Description
Acl	The number assigned to the ACL (a number from 1 to 99).
Clause	The number assigned to the clause (a number from 1 to 254).
Action	The method that incoming traffic is controlled by the clause (permit, deny, or bypass) and the protocol for the type of traffic.
Source	The configured source of the traffic.
Destination	The configured destination for the traffic.
Log	Whether or not ACL logging is enabled or disabled on the specified clause.
Content Hits	Increments for a packet received by the CSS before flow setup.
Router Hits	Increments for a packet directly forwarded to the CSS through a Telnet or FTP session or from non-TCP or UDP packets.
DNS Hits	Increments for a packet that matches an ACL clause for DNS flows.

Setting the Show ACL Counters to Zero

Use the **zero counts** command to set the content and DNS hit counters in the **show acl** command screen to zero for a specific ACL. You must be in an ACL to use this command. The CSS clears counters only for that ACL. The syntax and options for this command are:

```
(config-acl[7])# zero counts
```

ACL Example

The following ACL provides security for a CSS, Server1, and Server2 on one VLAN (VLAN1). The ACL:

- Permits clients from subnet 172.16.107.x to access servers 1 and 2 on VLAN1 using various applications (for example, Telnet, FTP, TFTP)
- Permits clients from subnet 172.16.107.x to launch a browser with the URL 172.16.107.35 (the Virtual IP address)
- Prevents clients on any subnet other than 172.16.107.x from accessing VLAN1 and servers 1 and 2

The individual clauses provide the following security.

- Clause 20 permits any protocol from source subnet 172.16.107.0 to Server1 (IP address 172.16.107.15).
- Clause 30 permits any protocol from source subnet 172.16.107.0 to Server2 (IP address 172.16.107.16).
- Clause 50 permits bidirectional communication to the VLAN for any ICMP traffic, including keepalives. If you are using service keepalives, you must configure a clause to permit keepalive traffic.
- Clause 60 permits UDP to port 520 on the VLAN for RIP updates. This clause is required if your router is on a subnet other than 172.16.107.x.
- Clause 70 denies everything that has not been permitted in the ACL.

```
!************************** ACL **************************
acl 1
clause 20 permit any 172.16.107.0 255.255.255.0 destination
172.16.107.15
clause 30 permit any 172.16.107.0 255.255.255.0 destination
172.16.107.16
```

```
clause 50 permit ICMP any destination any
clause 60 permit udp any eq 520 destination any
clause 70 deny any any destination any
apply circuit-(VLAN1)
```

Configuring Extension Qualifier Lists

An Extension Qualifier List (EQL) is a collection of file extensions that enable you to match a content rule based on extensions. You activate an EQL by associating it as part of a URL in a Layer 5 content rule. Use the **eql** command to access EQL configuration mode and configure an extension qualifier list. Enter a name that identifies the extension list you want to create. Enter an unquoted text string with no spaces and a length of 1 to 31 characters.

For example, enter:

```
(config)# eql graphics
(config-eql[graphics])#
```

To remove an existing EQL, use the **no eql** command from config mode. For example, enter:

```
(config)# no eql graphics
```

Once you create an EQL, you can configure the following attributes for it:

- **description** - Provides a description for the EQL. Enter a quoted text string with a maximum length of 64 characters. For example, enter:

  ```
  (config-eql[graphics])# description "This EQL specifies graphic
  file extensions"
  ```

- **extension** *name* - Specifies the extension *name* for content on which you want the CSS to match. Enter a text string from 1 to 7 characters. When configuring EQLs for services, make sure you enter an extension for static content such as .avi, .gif, or .jpg. Do not enter extensions for dynamic content such as .asp and .html. The order in which you enter extensions is irrelevant.

  For example, enter:

  ```
  (config-eql[graphics])# extension pcx
  ```

Optionally, you may provide a *description* of the extension type. Enter a quoted text string with a maximum length of 64 characters. For example, enter:

```
(config-eql[graphics])# extension gif "This is a graphics file"
```

To remove an extension from an EQL, use the **no extension** command. For example, enter:

```
(config-eql[graphics])# no extension gif
```

Specifying an Extension Qualifier List in a Uniform Resource Locator

Server selections are based on the Uniform Resource Locator (URL) specified in the owner content rule. To enable the CSS to access a service when a request for content matches the extensions contained in a previously defined Extension Qualifier List (EQL), specify the URL and EQL name for the content.

Specify a URL as a quoted text string with a maximum of 256 characters followed by **eql** and the EQL name.

Note Do not specify a file extension in the URL when you use an EQL in the URL or the CSS will return an error message. For example, the CSS will "return" an error message for the command **url "/*.txt" eql** graphics. The following command is valid; **url "/*" eql graphics**.

For example, enter:

```
(config-owner-content[arrowpoint.com-products.html])# url "/*" eql
graphics
```

The following example enables the CSS to direct all requests to the correct service for content that matches:

- Pathnames (*/customers/products*)
- Extensions listed in the EQL (*graphics*)

```
(config-owner-content[arrowpoint.com-products.html])# url
"/customers/products/*" eql graphics
```

To display an EQL name and extensions configured for a content rule, use the **show rule** command.

For details on the **show rule** command and its output, refer to Chapter 3, Configuring Content Rules.

Showing EQL Extensions and Descriptions

To display a list of existing EQLs names, use **eql ?** command.

For example, enter:

```
(config)# eql ?
```

To display the extensions configured for a specific EQL including any descriptions, use the **show eql** command and the EQL name. For example, enter:

```
(config)# show eql graphics
```

Table 5-6 describes the fields in the **show eql** output.

Table 5-6 Field Descriptions for the show eql Command

Field	Description
EQL	The name of the EQL and its description, if configured
Extensions	The extensions of content requests associated with the EQL and their descriptions, if configured

Configuring Uniform Resource Locator Qualifier Lists

URQL configuration mode allows you to configure a Uniform Resource Locator Qualifier List (URQL). A URQL is a group of URLs for content that you associate with one or more content rules. The CSS uses this list to identify which requests to send to a service. For example, you want all streaming video requests to be handled by your powerful servers. Create a URQL that contains the URLs for the content, and then associate the URQL to a content rule. The CSS will direct all requests for the streaming video URLs to the powerful servers specified in the content rule. Creating a URQL to group the URLs saves you from having to create a separate content rule for each URL.

Note You cannot specify both **url urql** and **application ssl** within the same content rule. You cannot configure a URQL with subscriber services.

Creating a URQL

To access URQL configuration mode, use the **urql** command. The prompt changes to (config-urql [name]). You can also use this command from URQL mode to access another URQL.

Enter the URQL name you want to create or enter an existing URQL. Enter the name as an unquoted text string with no spaces and a maximum of 31 characters. When you create a URQL, it remains suspended until you activate it using the **activate** command in urql mode. To display a list of existing URQL names, enter:

```
(config)# urql ?
```

For example, enter:

```
(config)# urql videos
(config-urql[videos)#
```

To remove an existing URQL, enter the following command in global configuration mode:

```
(config) no urql videos
```

Once you create a URQL:

1. Configure the URLs you want to group in the URQL by:
   a. Specifying the URL entry
   b. Defining the URL
   c. Optionally, describing the URL
2. Designate the domain name of the URLs in a URQL.
3. Add the URQL to a content rule using the owner-content **url** command.
4. Optionally, describe the URQL.

The following sections describe how to complete these tasks.

Configuring a URL in a URQL

Use the **url** command to include the URL for content requests you want as part of this URQL, and optionally provide a description. Configuring a URL in a URQL includes:

- Specifying the URL Entry
- Defining the URL
- Describing the URL

Note You must create the URL entry before you can define the URL, describe it, or associate it with a content rule.

Specifying the URL Entry

To specify a URL entry in a URQL, enter a URL number from 1 to 1000. For example, enter:

```
(config-urql[videos])# url 10
```

To remove a URL entry from a URQL, use the **no url** command. For example, enter:

```
(config-urql[videos])# no url 10
```

To specify additional URL entries in the URQL, reenter the **url** command. For example, enter:

```
(config-urql[videos])# url 20
(config-urql[videos])# url 30
(config-urql[videos])# url 40
```

Defining the URL

To define a URL for the entry, use the **url** command. Enter the URL as a quoted text string with a maximum of 251 characters. Wildcards are not allowed in a URQL URL. For example, enter:

```
(config-urql[videos])# url 10 url "/cooking/cookies.avi"
```

To remove a URL from an entry, use the **no url** *number* **url** command. Use this command to remove a previously assigned URL before you redefine the URL for an entry. For example, enter:

```
(config-urql[videos])# no url 10 url
```

To define additional URL for the entries, reenter the **url** *entry* **url** command. For example, enter:

```
(config-urql[videos])# url 20 url "/cooking/fudge.avi"
(config-urql[videos])# url 30 url "/cooking/pie.avi"
(config-urql[videos])# url 40 url "/cooking/cake.avi"
```

Describing the URL

You may optionally enter a description for the URL. Enter a quoted text string with a maximum length of 64 characters. For example, enter:

```
(config-urql[videos])# url 10 description "making cookies"
```

To remove a description about the URL, enter:

```
(config-urql[videos])# no url 10 description
```

Designating the Domain Name of URLs in a URQL

Use the **domain** command to designate the domain name or IP address of the URLs to a URQL. Enter the domain name in mnemonic host-name format (for example, www.arrowpoint.com) from 1 to 63 characters. Enter the IP address as a valid address for the domain name (for example, 192.168.11.1).

Note You must assign a domain before you can activate a URQL. To change the domain address of an existing URQL, suspend the URQL and then change the domain.

For example, enter:

```
(config-urql[videos])# domain "www.arrowpoint.com"
```

or

```
(config-urql[videos])# domain "192.168.11.1"
```

Adding a URQL to a Content Rule

Once you create and configure a URQL, use the **url urql** command to add it to a previously configured content rule. You can only assign one URQL per rule. Also, a content rule may contain either a URL or a URQL. To see a list of URQLs, use the **urql ?** command.

Note You cannot specify both **url urql** and **application ssl** within the same content rule. You cannot configure a URQL with subscriber services.

For example, enter:

```
(config-owner-content[chefsbest-recipes])# url urql videos
```

To remove a URQL from a content rule, enter:

```
(config-owner-content[chefsbest-recipes])# no url urql
```

To display a URL for a content rule, use the **show rule** command for the content rule. For details on the **show rule** command and its output, refer to Chapter 3, Configuring Content Rules.

Describing the URQL

Use the **description** command to provide a description for a URQL. Enter the description an a quoted text string with a maximum of 64 characters.

For example, enter:

```
(config-urql[videos])# description "cooking streaming video"
```

To clear a description for the URQL, enter:

```
(config-urql[videos])# no description
```

Activating a URQL

Use the **active** command to activate a suspended URQL. When you create a URQL, it is suspended until you use the **active** command to activate it.

Note Before you can activate a URQL, you must assign the domain for the URLs. See "Designating the Domain Name of URLs in a URQL" in this chapter.

For example, enter:

```
(config-urql[videos])# active
```

Suspending a URQL

Use the **suspend** command to deactivate a URQL on all currently assigned content rules. For example, enter:

```
(config-urql[videos])# suspend
```

To reactivate the URQL, use the **(config-urql) active** command.

URQL Configuration in a Startup-Config File

The following example shows a URQL configuration in a startup-config file.

```
!*************************** URQL ************************
urql excellence1
    url 10
    url 30
    url 30 url "/arrowpoint.gif"
    domain "192.168.128.109"
    url 10 url "/"
urql excellence2
    url 10
    url 10 url "/poweredby.gif"
    domain "192.168.128.109"
```

Showing URQLs

To display a list of URQLs, enter:

```
(config)# urql ?
```

To display all configured URQLs, enter:

```
(config)# show urql
```

To display a specific URQL, enter:

```
(config)# show urql videos
```

Table 5-7 describes the fields in the **show urql** output.

Table 5-7 Field Descriptions for the show urql Command

Field	Description
Name	The name of the URQL
Description	The configured description for the URQL
Domain	The domain name or address of the URLs associated with the URQL
Create Type	The create type (static or dynamic)

Table 5-7 Field Descriptions for the show urql Command (continued)

Field	Description
State	The state of the URQL (Active or Suspended)
Rules Associated	The number of rules associated with the URQL

Table 5-8 describes the additional fields when you display a specified URQL.

Table 5-8 Field Descriptions for a Specified URQL

Field	Description
URQL Table Domain	The domain name or address of the URLs associated with the URQL
Number of entries configured	The number of URL entries in the URQL
URL	The URL
Description	The description associated with the URL
Create Type	The create type (static or dynamic)
State	The state of the URL (Active or Suspended)
CSD Entries	The number of CSD entries

Configuring Network Qualifier Lists

NQL configuration mode allows you to configure a Network Qualifier List (NQL). An NQL is a list of networks or specific services, identified by IP address and subnet mask, that you assign to an ACL clause as a source or destination. By grouping networks into an NQL and assigning the NQL to an ACL clause, you have to create only one clause instead of a separate clause for each network.

The CSS enables you to configure a maximum of 512:

- Networks or services per NQL
- NQLs per CSS

This functionality is useful, for example, in a caching environment where you have a network you want to bypass and send content requests directly to the origin servers (servers containing the content). You can also use an NQL for users who prefer a service based on a specific network.

To access NQL configuration mode, use the **nql** command. The prompt changes to (config-nql [name]). You can also use this command from NQL mode to access another NQL.

See the following sections to configure an NQL:

- Creating an NQL
- Describing an NQL
- Adding Networks to an NQL
- Adding an NQL to an ACL Clause
- Showing NQL Configurations

Creating an NQL

Enter the name of the new NQL you want to create or an existing NQL. Enter the name as an unquoted text string with no spaces and a maximum of 31 characters. You can create a maximum of 512 NQLs per CSS.

For example, enter:

```
(config)# nql bypass_nql
(config-nql[bypass_nql])#
```

To display a list of existing NQLs, use the **nql ?** command. If no NQLs currently exist, the CSS prompts you to enter a new name.

To remove an existing NQL, use the **no nql** command. For example, enter:

```
(config)# no nql bypass_nql
```

Describing an NQL

Use the **description** command in NQL mode to provide a description for an NQL. Enter the NQL description as a quoted text string with a maximum length of 63 characters.

For example, enter:

```
(config-nql[bypass_nql])# description "Bypass services"
```

Adding Networks to an NQL

Use the **ip address** command to add a maximum of 512 networks or services to an NQL. Enter an IP address with either a subnet prefix or a subnet mask. You may also add an optional description for the IP address and turn on logging.

The syntax and options are:

ip address *ip_address*[*/subnet_prefix*| *subnet_mask*] {**"***description***"**}{**log**}

The variables and options are:

- *ip_address* - The destination network address. Enter the IP address in dotted-decimal notation (for example, 192.168.0.0).
- *subnet_mask* - The IP subnet mask prefix length in CIDR bitcount notation (for example, /16). The valid prefix length range is 8 to 32. Do not enter a space to separate the IP address from the prefix length.
- *subnet_address* - The IP subnet mask in dotted-decimal notation (for example, 255.255.0.0).
- **"***description***"** - A description of the IP address. Enter a quoted text string with a maximum of 63 characters.
- **log** - Logs an event involving an NQL. If you do not enter this option, events are not logged. To log an NQL event, you must enable global NQL logging. To enable global NQL logging, use the **(config) logging subsystem nql level debug-7** command. For logging information, refer to the *Cisco Content Services Switch Administration Guide*.

For example, to add two networks to the NQL *bypass_nql*, enter:

```
(config-nql[bypass_nql])# ip address 192.168.0.0/16 "Network of
dynamic mail content" log
(config-nql[bypass_nql])# ip address 123.123.123.0/24
```

To log events occurring on a network, you must also enable global NQL logging. For example, enter:

```
(config)# logging subsystem nql level debug-7
```

Note If you do not include a description or turn on logging when you create the entry and later wish to add a description or turn on logging, you must first remove the entry and then add it again with the desired options.

To remove an IP address from an NQL, use the **no ip address** command. For example, enter:

```
(config-nql[bypass_nql])# no ip address 192.168.0.0/16
```

Adding an NQL to an ACL Clause

To add an NQL to an ACL clause:

1. Create the ACL. For example, enter:

   ```
   (config)# acl 10
   ```

2. Define the clause, including the NQL as either a source or destination.

   This clause example bypasses content rules for any traffic from any source going to the destination networks defined in NQL *bypass_nql* on port 80.

   ```
   (config-acl[10])# clause 1 bypass any any destination nql
   bypass_nql eq 80
   ```

Showing NQL Configurations

Use the **show nql** command to display NQL configuration information. The syntax for this command is:

- **show nql** - Displays information for all NQLs. If you enter this command in NQL mode, the CSS displays the addresses only for the current NQL.
- **show nql** *nql_name* - Displays information for the specified NQL. Enter the NQL name as a case-sensitive unquoted text string with no spaces. To see a list of existing NQL names, use the **show nql ?** command.

For example, enter:

```
(config-nql[bypass_nql])# show nql
```

Table 5-9 describes the fields in the **show nql** output.

Table 5-9 Field Descriptions for the show nql Command

Field	Description
Name	The name of the NQL.
Description	The description associated with the NQL.
IP Addresses	The IP addresses and subnet mask supported by the NQL. If configured, a description appears after the address.

Configuring Domain Qualifier Lists

When you have a requirement for a content rule to match on multiple domain names, you can associate a Domain Qualifier List (DQL) to the rule. A DQL is a list of domain names that you configure and assign to a content rule, instead of creating a content rule for each domain. Assigning multiple domain names to a DQL enables you to have many domain names match on one content rule.

You can use a DQL on a rule to specify that content requests for each domain in the list will match on the rule. You can determine the order that the domain names are listed in the DQL. You can arrange the names in a DQL by assigning an index number as you add the name to the list.

Note The CSS supports a maximum of 512 DQLs, with a maximum of 2,500 DQL domain name entries. This means that a single DQL can have up to 2500 entries, or five DQLs can have up to 500 entries for each DQL.

DQLs exist independently of any range mapping. You can use them as a matching criteria to balance across servers that do not have VIP or port ranges. If you want to use range mapping when using range services, you need to consider the index of any domain name in the DQL. If you are not using service ranges with DQLs, you do not need to configure any index and the default index is 1.

For example, you could configure a DQL named Woodworker.

```
(config)# dql Woodworker
```

The domain names you could add as part of the DQL include *www.wood.com*, *www.woodworker.com*, *www.maple.com*, *www.oak.com*. You could configure *www.wood.com* and *www.woodworker.com* to have the same mapping index. You can enter indexes from 1 to 1000 and provide an optional quoted description for each index.

For example, enter:

```
(config-dql[Woodworker]# domain www.wood.com index 1 "This is the
same as the woodworker domain"
(config-dql[Woodworker]# domain www.woodworker.com index 1
(config-dql[Woodworker]# domain www.maple.com index 2
(config-dql[Woodworker]# domain www.oak.com index 3
```

If you specify a DQL as a matching criteria for content rule WoodSites, and there are two services, S1 and S2, associated with the rule, the CSS checks the services at mapping time for ranges. To add a DQL to a content rule, use the **url** command as shown:

```
(config-owner-content[WoodSites])# url "/*" dql Woodworker
```

For example, if the CSS receives a request for *www.oak.com* along with other criteria, a match on the WoodSites rule occurs on DQL index 3. If the rule has the roundrobin balance method configured, the CSS examines a service (S2 for this example) to determine the backend connection mapping parameters. If you configured S2 with a VIP address of 10.0.0.1 with a range of 5, the addresses include 10.0.0.1 through 10.0.0.5. Because this service has a range of address and **any** as its port, the DQL index of 3 matches the service VIP range index of 3, which is address 10.0.0.3.

To access DQL configuration mode, use the **dql** command from any configuration mode except boot, group, RMON alarm, RMON event, and RMON history configuration modes. The prompt changes to (config-dql [name]). You can also use this command from DQL mode to access an existing DQL.

See the following sections to configure a DQL:

- Creating a DQL
- Describing a DQL
- Adding a Domain to a DQL
- Adding a DQL to a Content Rule
- Removing a DQL from a Content Rule
- Showing DQL Configurations

Creating a DQL

To create a new DQL, enter the name of the DQL you want to create as an unquoted text string with no spaces and a maximum of 31 characters. To access an existing DQL, enter the DQL name. To display a list of existing DQL names, use the **dql ?** command.

For example, to configure a DQL:

```
(config)# dql pet_domains
(config-dql[pet_domains])#
```

Describing a DQL

Use the **description** command to provide a description for DQL. Enter the description as a quoted text string with a maximum of 63 characters, including spaces.

For example, enter:

```
(config-dql[pet_domains])# description "pet supplies"
```

Adding a Domain to a DQL

Use the **domain** command to add a domain to the list of domains supported by a DQL. The syntax is:

domain *name* **index** *number* {**"***description***"**}

The variables and option are:

- *name* - The name of the domain. Enter an unquoted text string with a maximum of 63 characters (for example, www.arrowpoint.com). The CSS matches the domain name exactly.
- *number* - The index number for the domain. Enter a number from 1 to 10000. If a domain has more than one domain name, you can assign the same index number to its different names.
- **"***description***"** - A description of the domain name. Enter a quoted text string with a maximum of 63 characters.

Note The CSS supports a maximum of 512 DQLs, with a maximum of 2,500 DQL domain name entries. This means that a single DQL can have up to 2500 entries, or five DQLs can have up to 500 entries for each DQL.

For example, enter:

```
(config-dql[pet_domains])# domain www.birds.com index 1
"idaho-based"
(config-dql[pet_domains])# domain www.cats.com index 2 "worldwide"
(config-dql[pet_domains])# domain www.horses.com index 3
"florida-based"
```

Normally, port 80 traffic does not use a port number in the domain name. To specify a port other than port 80, enter the domain name with the port number exactly. Separate the domain name and the port number with a colon. For example, enter:

```
(config-dql[pet_domains])# domain www.dogs.com:8080 index 4
```

To add or delete a domain name from a DQL that is assigned to a content rule, you must first suspend the content rule using the **suspend** command. You cannot make changes to a DQL currently in use by a content rule.

For example, to remove a domain from the example DQL, enter:

```
(config-dql[pet_domains])# no domain www.birds.com
```

Adding a DQL to a Content Rule

Once you have configured a DQL, use the **url** command to add it to a content rule. You cannot use wildcards in DQL entries.

For example, enter:

```
(config-owner-content[pets.com-rule1])# url "/*" dql pet_domains
```

Removing a DQL from a Content Rule

To remove a DQL that is assigned to a content rule, you must first suspend the content rule using the **suspend** command. You cannot remove a DQL currently in use by a content rule. Once the content rule is suspended, use the **no dql** command to remove the DQL from the content rule.

For example, enter:

```
(config) no dql pet_domains
```

Showing DQL Configurations

Use the **show dql** command to display all DQL configurations. To display a specific DQL, include the DQL name in the command line.

For example, enter:

```
(config-dql[pet_domains])# show dql pet_domains
```

Table 5-10 describes the fields in the **show dql** output.

Table 5-10 Field Descriptions for the show dql Command

Field	Description
Name	The name of the DQL
Index	The CSS unique index which identifies the DQL

Table 5-10 Field Descriptions for the show dql Command

Field	Description
Description	The description for the DQL
Index	The DQL unique index number for this domain
Domain	The name of the domain associated with the index number
Description	The description for the domain

Configuring Virtual Web Hosting

Virtual Web hosting enables you to host a large number of Web sites on a small number of servers (typically 2 to 10 servers) that have mirrored content. Each server may contain hundreds or thousands of Web sites. The servers determine which Web site is being requested based on IP address, port, and domain name.

Using virtual Web hosting, you may configure:

- Services with either a range of IP addresses or a range of ports.
- Content rules with either a range of VIPs or a DQL (but not both). This would allow the CSS to map the range of VIPs or the domain names in the DQL to the servers.
- Content rules with either a range of VIPS or a DQL (but not both) that would map to a server without a range. This allows the CSS to map many domain names to one server.

You can configure the CSS to load balance the Web sites by configuring port ranges, VIP ranges, or DQLs. For more information on the service and content rule commands required, see Chapter 1, Configuring Services and Chapter 3, Configuring Content Rules.

See Table 5-11 for the steps required to configure virtual Web hosting.

Table 5-11 Virtual Web Hosting Configuration Quick Start

Task and Command Example
1. Enter config mode by typing **config**. `(config)#`

2. Create a service.
`(config)# service serv1`
`(config-service[serv1])#`

3. Assign an IP address to the service and define the IP address range. Enter a number from 1 to 65535. When using the **ip address range** command, use IP addresses that are within the subnet you are using. The CSS does not use ARP for IP addresses that are not on the circuit subnet. `(config-service[serv1])# ip address 10.3.6.1 range 200`

4. Configure other service rules as needed (for example, protocol, keepalive parameters).
`(config-service[serv1])# protocol tcp`
`(config-service[serv1])# keepalive type http`
`(config-service[serv1])# keepalive method get`
`(config-service[serv1])# keepalive uri "/index.html"`

5. Activate the service. `(config-service[serv1])# active`

6. Create the content rule.
`(config-owner[arrowpoint])# content rule1`
`(config-owner-content[arrowpoint-rule1])#`

7. Configure a VIP. You can define a VIP range only if you do not plan to configure a DQL. `(config-owner-content[arrowpoint-rule1])# vip address 192.168.3.6 range 10` When using the **vip address range** command, use IP addresses that are within the subnet you are using. The CSS does not use ARP for IP addresses that are not on the circuit subnet.

Table 5-11 Virtual Web Hosting Configuration Quick Start (continued)

Task and Command Example

8. Configure other content rule commands as needed (for example, port, protocol, and add a service).
`(config-owner-content[arrowpoint-rule1])# port 80`
`(config-owner-content[arrowpoint-rule1])# protocol tcp`
`(config-owner-content[arrowpoint-rule1])# add service serv1`

9. Activate the content rule. `(config-owner-content[arrowpoint-rule1])# active`

10. If you have not configured a VIP range, you can create a DQL.
`(config)# dql pet_domains`
`(config-dql[pet_domains])#`

11. Add domains to the DQL you created.
`(config-dql[pet_domains])# domain www.birds.com index 1 "idaho-based"`
`(config-dql[pet_domains])# domain www.cats.com index 2 "worldwide"`
`(config-dql[pet_domains])# domain www.horses.com index 3 "florida-based"`

12. Add the DQL to the content rule using the **url** command. `(config-owner-content[arrowpoint-rule1])# url "/*" dql pet_domains`

Where to Go Next

You can configure HTTP header load balancing by creating an HTTP header field group and configuring HTTP header fields. For information, see Chapter 6, Configuring HTTP Header Load Balancing.