Configuring the Content Switching Module

This chapter describes how to configure the CSM and contains these sections:

Preparing to Configure the CSM

Before you configure the CSM, you must take these actions:

- Be sure that the Cisco IOS versions for the switch and the module match. Refer to the "Software Compatibility" section on page 2-4.
- Before you can configure server load balancing, you must obtain the following information:
  - Network topology that you are using in your installation
  - Real server IP addresses
  - An entry for the CSM VIPs in the Domain Name Server (DNS) (if you want them to be reached through names)
  - Each virtual server's IP address

- You must configure VLANs on the Catalyst 6500 series switch before you configure VLANs for the CSM. VLAN IDs must be the same for the switch and the module. Refer to the *Catalyst 6500 Series Software Configuration Guide* for details.

  This example shows how to configure VLANs:

```
Router>
Router> enable
Router# vlan database
Router(vlan)# vlan 130
VLAN 130 added:
    Name: VLAN130
Router(vlan)# vlan 150
VLAN 150 added:
    Name: VLAN150
Router(vlan)# exit
```

- Place physical interfaces that connect to the servers or to the clients in the corresponding VLAN.

  This example shows how to configure a physical interface as a Layer 2 interface and assign it to a VLAN:

```
Router>
Router> enable
Router# config
Router(config)# interface 3/1
Router(config-if)# switchport
Router(config-if)# switchport access vlan 150
Router(config-if)# no shutdown
Router(vlan)# exit
```

- If the Multilayer Switch Function Card (MSFC) is used on the next-hop router on either the client or the server-side VLAN, then you must configure the corresponding Layer 3 VLAN interface.

Caution You cannot use the MSFC simultaneously as the router for both the client and the server side unless Policy Based Routing is used and the CSM is configured in router mode. If you use the CSM in bridge (single subnet) mode, do not configure the Layer 3 VLAN interface on the MSFC for both the client and the server side. If you use the CSM in router mode, do not configure the Layer 3 VLAN interface on the MSFC for both the client and the server side unless you properly configure Policy Based Routing to direct return traffic back to the CSM.

This example shows how to configure the Layer 3 VLAN interface:

```
Router>
Router> enable
Router# config
Router(config)# interface vlan 130
Router(config-if)# ip address 10.10.1.10 255.255.255.0
Router(config-if)# no shutdown
Router(vlan)# exit
```

Using the Command-Line Interface

The software interface for the CSM is the Cisco IOS command-line interface. To understand the Cisco IOS command-line interface and Cisco IOS command modes, refer to Chapter 2 in the *Catalyst 6500 Series IOS Software Configuration Guide*.

Note Because of each prompt's character limit, some prompts may be truncated. For example: Router(config-slb-vlan-server)# may appear as Router(config-slb-vlan-serve)#.

Accessing Online Help

In any command mode, you can get a list of available commands by entering a question mark (?) as follows:

```
Router> ?
```

or

```
Router(config)# module csm 5
Router(config-module-csm)# ?
```

Note Online help shows the default configuration values and ranges available to commands.

Upgrading to a New Software Release

This section describes three methods for upgrading the CSM:

- Upgrading from the Supervisor Engine Bootflash, page 3-4
- Upgrading from a PCMCIA Card, page 3-4
- Upgrading from an External TFTP Server, page 3-5

Note When upgrading to a new software release, you must upgrade the CSM image before upgrading the Cisco IOS image. Failure to do so causes the supervisor engine not to recognize the CSM. In this case, you would have to downgrade the Cisco IOS image, upgrade the CSM image, and then upgrade the Cisco IOS image.

To upgrade the CSM you need to session into the CSM module being upgraded. During the upgrade, enter all commands on a console connected to the supervisor engine. Enter each configuration command on a separate line. To complete the upgrade, enter the **exit** command to return to the supervisor engine prompt.

Caution You must enter the **exit** command to terminate sessions with the CSM that is being upgraded. If you do not terminate the session and you remove the CSM from the Catalyst 6500 series chassis, you cannot enter configuration commands to the CSM unless you press **Ctrl** + **^**, enter **x**, and enter the **disconnect** command at the prompt.

Upgrading from the Supervisor Engine Bootflash

Note Refer to the *Catalyst 6500 Series Supervisor Engine Flash PC Card Installation Note* for instructions on loading images into bootflash.

To upgrade the CSM from the supervisor engine bootflash, perform these steps:

Step 1 Enable the TFTP server to supply the image from bootflash as follows:

```
Router>
Router> enable
Router# configure terminal
Router(config)# tftp-server sup-bootflash:c6slb-apc.revision-num.bin
Router(config)
```

Step 2 Set up a session between the supervisor engine and the CSM:

```
Router# session slot csm-slot-number processor 0
```

Step 3 Load the image from the supervisor engine to the CSM:

```
CSM> upgrade 127.0.0.zz c6slb-apc.revision-num.bin
```

where:

zz = 12 if the supervisor engine is installed in chassis slot 1.
zz = 22 if the supervisor engine is installed in chassis slot 2.

Note The supervisor engine only can be installed in chassis slot 1 or slot 2.

Step 4 Close the session to the CSM, and return to the IOS prompt:

```
CSM> exit
```

Step 5 Reboot the CSM by power cycling the CSM or by entering the following commands on the supervisor engine console:

```
Router# hw-module module csm-slot-number reset
```

Upgrading from a PCMCIA Card

Note Throughout this publication, the term *Flash PC card* is used in place of the term *PCMCIA card*.

To upgrade the CSM from a removable Flash PC card inserted in the supervisor engine, perform these steps:

Step 1 Enable the TFTP server to supply the image from the removable Flash PC card:

```
Router>
Router> enable
Router# configure terminal
```

```
Router(config)# tftp-server slotx:c6slb-apc.revision-num.bin
```

where:

x = 0 if the Flash PC card is installed in supervisor engine PCMCIA slot 0.

Step 2 Set up a session between the supervisor engine and the CSM:

```
Router# session slot csm-slot-number processor 0
```

Step 3 Load the image from the supervisor engine to the CSM:

```
CSM> upgrade slot0: c6slb-apc.revision-num.bin
```

Note The supervisor engine can only be installed in chassis slot 1 or slot 2.

Step 4 Close the session to the CSM and return to the IOS prompt:

```
CSM> exit
```

Step 5 Reboot the CSM by power cycling the CSM or by entering the following commands on the supervisor engine console:

```
Router# hw-module module csm-slot-number reset
```

Upgrading from an External TFTP Server

To upgrade the CSM from an external TFTP server, perform these steps:

Step 1 Create a VLAN on the supervisor engine for the TFTP CSM runtime image download.

Note You can use an existing VLAN, however, for a reliable download, you should create a VLAN specifically for the TFTP connection.

Step 2 Configure the interface that is connected to your TFTP server.

Step 3 Add the interface to the VLAN.

Step 4 Enter the CSM **vlan** command. See the "Configuring VLANs" section on page 3-10 for more information.

Step 5 Add an IP address to the VLAN for the CSM.

Step 6 Enter the **show csm** *slot* **vlan detail** command to verify your configuration. See the "Configuring VLANs" section on page 3-10 for more information.

Step 7 Verify the CSM connectivity to the TFTP server:

```
Router# ping module csm csm-slot-number TFTP-server-IP-address
```

Step 8 Set up a session between the supervisor engine and the CSM:

```
Router# session slot csm-slot-number processor 0
```

Step 9 Upgrade the image

```
CSM> upgrade TFTP-server-IP-address c6slb-apc.rev-number.bin
```

Step 10 Close the session to the CSM and return to the IOS prompt:

```
CSM> exit
```

Step 11 Reboot the CSM by power cycling the CSM or by entering the following commands on the supervisor engine console:

```
Router# hw-module module csm-slot-number reset
```

Saving and Restoring Configurations

For information about saving and restoring configurations, refer to the *Catalyst 6500 Series IOS Software Configuration Guide*.

Configuring CSM Modes

Load balancing on the Catalyst 6500 series switch can operate in two modes: the routed processor (RP) mode and the CSM mode. By default, the CSM is configured in RP mode. The RP mode allows you to configure one or multiple CSMs in the same chassis and run Cisco IOS SLB on the same switch.

Note The RP mode is not only the default mode but it is recommended. The CSM mode is only kept for backward compatibility with CSM software images previous to 2.1.

The following sections provide information about CSM modes:

- Specifying CSM Locations, page 3-6
- Mode Command Syntax, page 3-7
- Migrating Between Modes, page 3-7

CSM mode allows you to configure a single CSM only. The CSM mode is supported for backward compatibility with previous software releases. The single CSM configuration will not allow Cisco IOS SLB to run on the same switch.

Specifying CSM Locations

Before you can enter CSM configuration commands on the switch, you must specify the CSM that you want to configure. To specify a CSM for configuration, use the **module csm** *slot-number* command. The *slot-number* is the chassis slot where the CSM being configured is located.

The **module csm** command places you in CSM configuration submode. All further configuration commands that you enter apply to the CSM installed in the slot you have specified.

Note Unless otherwise specified, all the examples in this publication assume that you have already entered this command and entered the configuration submode for the CSM you are configuring.

Mode Command Syntax

The command syntax for CSM mode and RP mode configuration is identical with these exceptions:

- When configuring in CSM mode, you must prefix each top-level command with **ip slb**.
- Prompts are different for CSM mode and for RP mode configurations.

To configure a virtual server for multiple CSMs, perform this task:

	Command	Purpose
Step 1	`Router(config)# module csm 5`	Specifies the location of the CSM you are configuring.
Step 2	`Router(config-module-csm)# vserver VS1`	Configures the virtual server.

Migrating Between Modes

Existing CSM configurations are migrated to the new configuration when the mode is changed from **csm** to **rp** using the **ip slb mode** command. If any Cisco IOS SLB or CSM configuration exists, you are prompted for the slot number.

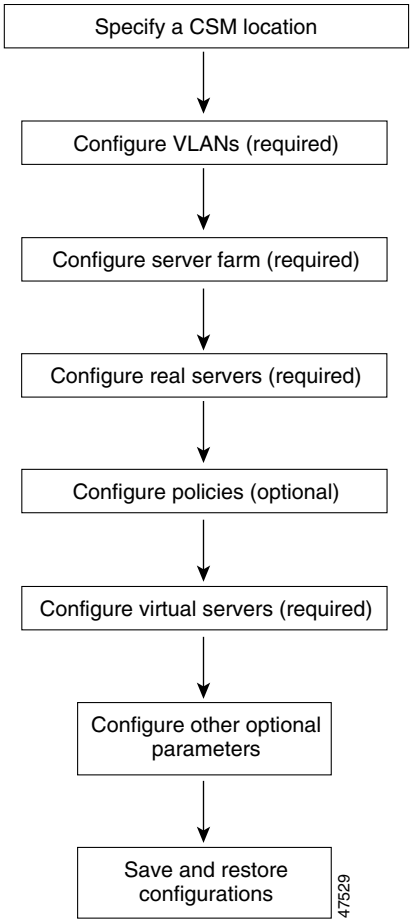
You can migrate from an RP mode configuration to CSM mode configuration on the Catalyst 6500 series switch. You can only manually migrate from a Cisco IOS SLB configuration to a CSM configuration.

Configuration Overview

The configuration process described here assumes that the switch is in the RP mode. Figure 3-1 shows an overview of the required and optional operations in the configuration process.

Note Configuring policies is not necessary for Layer 4 load balancing.

Figure 3-1 Configuration Overview



To configure the required parameters, see the following sections:

- Configuring VLANs, page 3-10
- Configuring Server Farms, page 3-12
- Configuring Real Servers, page 3-13
- Configuring Policies, page 3-14
- Configuring Virtual Servers, page 3-18

After you configure the required load-balancing parameters on the CSM, you can configure the optional parameters in the following sections:

- Configuring TCP Parameters, page 3-21
- Configuring Dynamic Feedback Protocol, page 3-21
- Configuring Redirect Virtual Servers, page 3-22
- Configuring Client NAT Pools, page 3-23
- Configuring Server-Initiated Connections, page 3-24

To save or restore your configurations or to work with advanced configurations, refer to the following sections in Chapter 3 through Chapter 6:

- Saving and Restoring Configurations, page 3-6
- Configuring the Single Subnet (Bridge) Mode, page 4-2
- Configuring the Secure (Router) Mode, page 4-4
- Configuring Fault Tolerance, page 4-5
- Configuring HSRP, page 4-9
- Configuring URL Hashing, page 5-1
- Configuring Firewall Load Balancing, page 5-3
- Configuring Generic Header Parsing, page 5-27
- Configuring Persistent Connections, page 5-30
- Configuring Connection Redundancy, page 5-30
- Configuring a Hitless Upgrade, page 5-31
- Configuring SNMP Traps for Real Servers, page 5-32
- Configuring TCL Scripts, page 5-37
- Configuring Probes for Health Monitoring, page 6-1
- Configuring Route Health Injection, page 6-6
- Configuring Inband Health Monitoring, page 6-10
- Configuring HTTP Return Code Checking, page 6-11
- Configuring Scripts for Health Monitoring Probes, page 6-12

Configuring VLANs

When you install the CSM in a Catalyst 6500 series switch, you need to configure client-side and server-side VLANs. (See Figure 3-2.)

Note You must configure VLANs on the Catalyst 6500 series switch before you configure VLANs for the CSM. VLAN IDs must be the same for the switch and the module.

Figure 3-2 Configuring VLANs

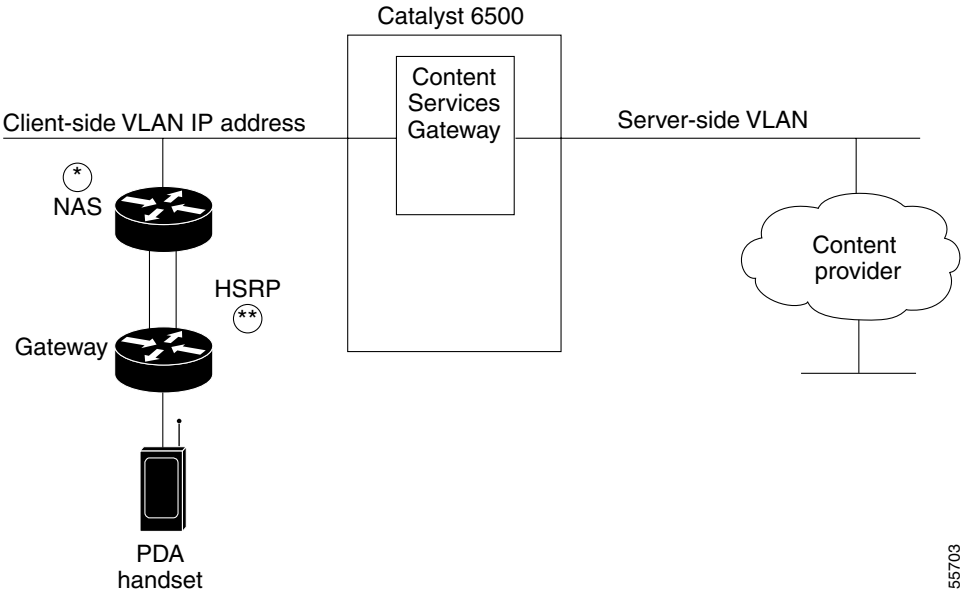


Diagram notes:

*Any router configured as a client-side gateway or a next-hop router for servers more than one hop away must have ICMP redirects disabled. The CSM does not perform a Layer 3 lookup to forward traffic; the CSM cannot act upon ICMP redirects.

** You can configure up to seven gateways per VLAN for up to 256 VLANs and up to 224 gateways for the entire system. If an HSRP gateway is configured, the CSM uses 3 gateway entries out of the 224 gateway entries because traffic can come from the virtual and physical MAC addresses of the HSRP group. (See the "Configuring HSRP" section on page 4-9.)

Configuring Client-Side VLANs

To configure client-side VLANs, perform this task:

Caution You cannot use VLAN 1 as a client-side or server-side VLAN for the CSM.

	Command	Purpose
Step 1	`Router(config-module-csm)# vlan vlanid client`	Configures the client-side VLANs and enters the client VLAN mode[1].
Step 2	`Router(config-slb-vlan-client)# ip ip-address netmask`	Configures an IP address to the CSM used by probes and ARP requests on this particular VLAN[2].
Step 3	`Router(config-slb-vlan-client)# gateway ip-address`	Configures the gateway IP address.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example shows how to configure the CSM for client-side VLANs:

```
Router(config-module-csm)# vlan 130 client
Router(config-slb-vlan-client)# ip addr 123.44.50.6 255.255.255.0
Router(config-slb-vlan-client)# gateway 123.44.50.1
Router(config-slb-vlan-client)# exit
Router# show module csm vlan 1
```

Configuring Server-Side VLANs

To configure server-side VLANs, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# vlan vlanid server`	Configures the server-side VLANs and enters the server VLAN mode[1].
Step 2	`Router(config-slb-vlan-server)# ip ip-address netmask`	Configures an IP address for the server VLAN[2].
Step 3	`Router(config-slb-vlan-server)# alias ip-address netmask`	(Optional) Configures multiple IP addresses to the CSM as alternate gateways for the real server[3].
Step 4	`Router(config-slb-vlan-server)# route ip-address netmask gateway gw-ip-address`	Configures a static route to reach the real servers if they are more than one Layer 3 hop away from the CSM.
Step 5	`Router # show module csm slot vlan [client | server | ft] [id vlan-id] [detail]`	Displays the client-side and server-side VLAN configurations.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.
3. The alias is required in the redundant configuration. (See the "Configuring Fault Tolerance" section on page 4-5.)

This example shows how to configure the CSM for server-side VLANs:

```
Router(config-module-csm)# vlan 150 server
Router(config-slb-vlan-server)# ip addr 123.46.50.6 255.255.255.0
Router(config-slb-vlan-server)# alias 123.60.7.6 255.255.255.0
Router(config-slb-vlan-server)# route 123.50.0.0 255.255.0.0 gateway 123.44.50.1
Router(config-slb-vlan-server)# exit
```

Configuring Server Farms

A server farm or server pool is a collection of servers that contain the same content. You specify the server farm name when you configure the server farm and add servers to it, and when you bind the server farm to a virtual server. When you configure server farms, do the following:

- Name the server farm.
- Configure a load-balancing algorithm (predictor) and other attributes of the farm
- Set or specify a set of real servers. (See the "Configuring Real Servers" section on page 3-13.)
- Set or specify the attributes of the real servers.

You also can configure inband health monitoring for each server farm. (See the "Configuring Inband Health Monitoring" section on page 6-10.) You can assign a return code map to a server farm to configure return code parsing. (See the "Configuring HTTP Return Code Checking" section on page 6-11.)

To configure server farms, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# serverfarm serverfarm-name`	Creates and names a server farm and enters the server farm configuration mode[1] [2].
Step 2	`Router(config-slb-sfarm)# predictor [roundrobin | leastconns | hash url | hash address [source | destination] [ip-netmask] | forward]]`	Configures the load-balancing prediction algorithm[2]. If not specified, the default is **roundrobin**.
Step 3	`Router(config-slb-sfarm)# nat client client-pool-name`	(Optional) Enables the NAT mode, client[2]. (See the "Configuring Client NAT Pools" section on page 3-23.)
Step 4	`Router(config-slb-sfarm)# no nat server`	(Optional) Specifies that the destination IP address is not changed when the load balancing decision is made.
Step 5	`Router(config-slb-sfarm)# probe probe-name`	(Optional) Associates the server farm to a probe that can be defined by the **probe** command[2].
Step 6	`Router(config-slb-sfarm)# bindid bind-id`	(Optional) Binds a single physical server to multiple server farms and reports a different weight for each one[2]. The **bindid** is used by DFP.
Step 7	`Router(config-slb-sfarm)# failaction purge`	(Optional) Sets the behavior of connections to real servers that have failed[2].
Step 8	`Router(config-slb-real)# inservice`	Enables the real servers.
Step 9	`Router# show module csm slot serverfarm serverfarm-name [detail]`	Displays information about one or all server farms.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example shows how to configure a server farm, named p1_nat, using the least-connections (**leastconns**) algorithm. The real server with the fewest number of active connections will get the next connection request for the server farm with the leastconns predictor.

```
Router(config-module-csm)# serverfarm pl_nat
Router(config-slb-sfarm)# predictor leastconns
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.1.0.106
Router(config-slb-sfarm)# inservice
```

Configuring Real Servers

Real servers are physical devices assigned to a server farm. Real servers provide the services that are load balanced. When the server receives a client request, it sends the reply to the CSM for forwarding to the client.

You configure the real server in the real server configuration mode by specifying the server IP address and port when you assign it to a server farm. You enter the real server configuration mode from the server farm mode where you are adding the real server.

To configure real servers, perform this task:

	Command	Purpose
Step 1	`Router(config-slb-sfarm)# real ip-address [port]`	Identifies a real server as a member of the server farm and enters the **real** server configuration mode. An optional translation port can also be configured[1, 2].
Step 2	`Router(config-slb-real)# weight weighting-value`	(Optional) Sets the weighting value for the virtual server predictor algorithm to assign the server's workload capacity relative to the other servers in the server farm if the round robin or least connection is selected[2].
Step 3	`Router(config-slb-real)# maxconns max-conns`	(Optional) Sets the maximum number of active connections on the real server[2]. When the specified maximum is reached, no more new connections are sent to that real server until the number of active connections drops below the minimum threshold.
Step 4	`Router(config-slb-real)# minconns min-conns`	(Optional) Sets the minimum connection threshold[2].
Step 5	`Router(config-slb-real)# inservice`	Enables the real server for use by the CSM[2 3].
Step 6	`Router# show module csm slot [sfarm serverfarm-name] [detail]`	(Optional) Displays information about configured real servers. The **sfarm** option limits the display to real servers associated with a particular virtual server. The **detail** option displays detailed real server information.
Step 7	`Router# show module csm slot [vserver virtserver-name] [client ip-address] [detail]`	Displays active connections to the CSM. The **vserver** option limits the display to connections associated with a particular virtual server. The **client** option limits the display to connections for a particular client. The **detail** option displays detailed connection information.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.
3. Repeat Steps 1 through 5 for each real server you are configuring.

This example shows how to create real servers:

```
Router(config-module-csm)# serverfarm serverfarm
Router(config-slb-sfarm)# real 10.8.0.7
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.8.0.8
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.8.0.9
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.8.0.10
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.1.0.106
Router(config-slb-sfarm)# inservice
Router(config-slb-real)# end
Router# show reals detail
Router# show conns detail
```

Configuring Policies

Policies are access rules that traffic must match when balancing to a server farm. Policies allow the CSM to balance Layer 7 traffic. Multiple policies can be assigned to one virtual server, creating multiple access rules for that virtual server. When configuring policies, you first configure the access rules (maps, client-groups, and sticky groups) and then you combine these access rules under a particular policy.

Note You must associate a server farm with a policy. A policy that does not have an associated server farm cannot forward traffic. The server farm associated with a policy receives all the requests that match that policy.

When the CSM is able to match policies, it selects the policy that appears first in the policy list. Policies are located in the policy list in the sequence in which they were bound to the virtual server. You can reorder the policies in the list by removing policies and reentering them in the correct order. Enter the **no slb-policy** *policy name* command and the **slb-policy** *policy name* command in the *vserver* submode to remove and enter policies.

To configure load-balancing policies, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# policy policy-name`	Creates the policy and enters the policy submode to configure the policy attributes[1].
Step 2	`Router(config-slb-policy)# url-map url-map-name`	Associates a URL map to a policy[2]. You must have previously created and configured the URL maps and cookie maps with the **map** command. See the "Configuring Maps" section on page 3-15.
Step 3	`Router(config-slb-policy)# cookie-map cookie-map-name`	Associates a cookie map to a policy[2].
Step 4	`Router(config-slb-policy)# header-map name`	Associates an HTTP header map to a policy.
Step 5	`Router(config-slb-policy)# sticky-group group-id`	Associates this policy to a specific sticky group[2].

	Command	Purpose
Step 6	`Router(config-slb-policy)# client-group value | std-access-list-name`	Configures a client filter associated with a policy. Only standard IP access lists are used to define a client filter.
Step 7	`Router(config-slb-policy)# serverfarm serverfarm-name`	Configures the server farm serving a particular load-balancing policy. Only one server farm can be configured per policy[2].
Step 8	`Router(config-slb-policy)# set ip dscp dscp-value`	Marks traffic with a *dscp-value* if packets matched with the load-balancing policy[2].

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example assumes that the URL map, **map1**, has already been configured and shows how to configure server load-balancing policies and associate them to virtual servers:

```
Router(config-slb-policy)# serverfarm pl_sticky
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-sfarm)# inservice
Router(config-slb-policy)# exit
Router(config-module-csm)# policy policy_sticky_ck
Router(config-slb-policy)# serverfarm pl_sticky
Router(config-slb-policy)# url-map map1
Router(config-slb-policy)# exit
Router(config-module-csm)# vserver vs_sticky_ck
Router(config-slb-vserver)# virtual 10.1.0.80 tcp 80
Router(config-slb-vserver)# slb-policy policy_sticky_ck
Router(config-slb-sfarm)# inservice
Router(config-slb-policy)# exit
```

Configuring Maps

You configure maps to define multiple URLs, cookies, HTTP headers, and return codes into groups that can be associated with a policy when you configure the policy. (See the "Configuring Policies" section on page 3-14.) Regular expressions for URLs (for example, *url1* and *url2*) are based on UNIX filename specifications. See Table 3-1 for more information.

To add a URL map, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# map url-map-name url`	Creates a group to hold multiple URL match criteria.[1, 2]
Step 2	`Router(config-slb-map-url)# match protocol http url url-path`	Specifies a string expression to match against the requested URL[2].

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

Table 3-1 Special Characters for Matching String Expressions

Convention	Description
*	Zero or more characters.
?	Exactly one character. **Note** You must precede the question mark with a Ctrl-V command to prevent the CLI Parser from interpretingit as a help request
\	Escaped character.
Bracketed range [0-9]	Matching any single character from the range.
A leading ^ in a range	Do not match any in the range. All other characters represent themselves.
.\a	Alert (ASCII 7).
.\b	Backspace (ASCII 8).
.\f	Form-feed (ASCII 12).
.\n	New line (ascii 10).
.\r	Carriage return (ASCII 13).
.\t	Tab (ASCII 9).
.\v	Vertical tab (ASCII 11).
.\0	Null (ASCII 0).
.\\	Backslash.
.\x##	Any ASCII character as specified in two-digit hex notation.

To add a cookie map, perform this task:

	Command	Purpose
Step 1	`Router(config)# map` *cookie-map-name* `cookie`	Configures multiple cookies into a cookie map[1].
Step 2	`Router(config-slb-map-cookie)# match protocol http cookie` *cookie-name* `cookie-value` *cookie-value-expression*	Configures multiple cookies[1].

1. The **no** form of this command restores the defaults.

This example shows how to configure maps and associate them with a policy:

```
Router(config-module-csm)# serverfarm pl_url_url_1
Router(config-slb-sfarm)# real 10.8.0.26
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-slb-policy)# serverfarm pl_url_url_1
Router(config-slb-policy)# url-map url_1
Router(config-slb-policy)# exit
Router(config-module-csm)# serverfarm pl_url_url_2
Router(config-slb-sfarm)# real 10.8.0.27
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
```

```
Router(config-slb-sfarm)# exit
Router(config-module-csm)# map url_1  url
Router(config-slb-map-url)# match protocol http url /url1
Router(config-slb-map-url)# exit
Router(config-module-csm)# map url_2 url
Router(config-slb-map-url)# match protocol http url /url/url/url
Router(config-slb-map-url)# match protocol http url /reg/*long.*
Router(config-slb-map-url)# exit
Router(config-module-csm)# policy policy_url_1
Router(config-module-csm)# policy policy_url_2
Router(config-slb-policy)# serverfarm pl_url_url_2
Router(config-slb-policy)# url-map url_2
Router(config-slb-policy)# exit
Router(config-module-csm)# vserver vs_url_url
Router(config-slb-vserver)# virtual 10.8.0.145 tcp 80
Router(config-slb-vserver)# slb-policy policy_url_1
Router(config-slb-vserver)# slb-policy policy_url_2
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
```

Using the **map** command, you create a map group with the type HTTP header. Entering the **map** command places you in a submode where you can specify the header fields and values for CSM to search for in the request.

To create a map for the HTTP header, perform this task:

Command	Purpose
`Router(config-module-csm)# map name header`	Creates and names an HTTP header map group.

For more information about header maps, see the "Configuring Generic Header Parsing" section on page 5-27.

To create a map for return code checking, perform this task:

Command	Purpose
`Router(config-module-csm)# map name retcode`	Creates and names a return code map group.

For more information about return code maps, see the "Configuring HTTP Return Code Checking" section on page 6-11.

Configuring Sticky Groups

Configuring a sticky group involves configuring the attributes of that group and associating it with a policy. Sticky time specifies the period of time that the sticky information is kept. The default sticky time value is 1440 minutes (24 hours).

To configure sticky groups, perform this task:

Command	Purpose
`Router(config-module-csm)# sticky sticky-group-id [netmask netmask | cookie name | ssl] [timeout sticky-time] [source | destination | both]`	Ensures that connections from the same client matching the same policy use the same real server[1].

1. The **no** form of this command restores the defaults.

This example shows how to configure a sticky group and associate it with a policy:

```
Router(config-module-csm)# sticky 1 cookie foo timeout 100
Router(config-module-csm)# serverfarm pl_stick
Router(config-slb-sfarm)# real 10.8.0.18
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.8.0.19
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-module-csm)# policy policy_sticky_ck
Router(config-slb-policy)# serverfarm pl_stick
Router(config-slb-policy)# sticky-group 1
Router(config-slb-policy)# exit
Router(config-module-csm)# vserver vs_sticky_ck
Router(config-slb-vserver)# virtual 10.8.0.125 tcp 90
Router(config-slb-vserver)# slb-policy policy_sticky_ck
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
```

Configuring Virtual Servers

Virtual servers represent groups of real servers and are associated with real server farms through policies. Configuring virtual servers requires that you set the attributes of the virtual server specifying the default server farm (default policy) and that you associate other server farms through a list of policies. The default server farm (default policy) is used if a request does not match any SLB policy or if there are no policies associated with the virtual server.

Before you can associate a server farm with the virtual server, you must configure the server farm. For more information, see the "Configuring Server Farms" section on page 3-12. Policies are processed in the order in which they are entered in the virtual server configuration. For more information, see the "Configuring Policies" section on page 3-14.

You can configure each virtual server with a pending connection timeout to terminate connections quickly if the switch becomes flooded with traffic. This connection applies to a transaction between the client and server that has not completed the request and reply process.

In a service provider environment in which different customers are assigned different virtual servers you may need to balance the connections to prevent an individual server from absorbing most or even all of the connection resources on the CSM. You can limit the number of connections going through the CSM to a particular virtual server by using the VIP connection watermarks feature. With this feature, you may set limits on each virtual server, allowing a fair distribution of connection resources among all virtual servers.

Note You can configure a single virtual server to operate at either Level 4 or Level 7. To configure a virtual server to operate at Level 4, specify the server farm (default policy) as part of the virtual server configuration. (See Step 3 in the following task table.) To configure a virtual server to operate at Level 7, add SLB policies in the configuration of the virtual server. (See Step 7 in the following task table.)

The CSM can load-balance traffic from any IP protocol. When you configure a virtual server in *vserver* submode, you must define the IP protocol that the virtual server will accept.

Note Although all IP protocols have a protocol number, the CSM allows you to specify TCP or UDP by name instead of requiring you to enter their numbers.

Configure the virtual server in the virtual server configuration submode.

To configure virtual servers, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# owner` *owner-name* **address** *street-address-information* **billing-info** *billing-address-information* **email-address** *email-information* **maxconns** *1:MAXULONG*	Restricts access to virtual servers to a specific owner object.
Step 2	`Router(config-module-csm)# vserver` *virtserver-name*	Identifies the virtual server and enters the virtual server configuration mode[1, 2].
Step 3	`Router(config-slb-vserver)# vs-owner` *owner-name* **maxconns** *1:MAXULONG*	Sets the owner object name for this virtual server.
Step 4	`Router(config-slb-vserver)# virtual` *ip-address [ip-mask] protocol port-number* [**service ftp**]	Sets the IP address for the virtual server optional port number or name and the connection coupling and type[2]. The *protocol* value is **tcp**, **udp**, **Any** (no port-number is required), or a *number* value (no port-number is required).
Step 5	`Router(config-slb-vserver)# serverfarm` *serverfarm-name*	Associates the default server farm with the virtual server[2 3]. Only one server farm is allowed. If the server farm is not specified, all the requests not matching any other policies will be discarded.
Step 6	`Router(config-slb-vserver)# sticky` *duration*	(Optional) Configures connections from the client to use the same real server[2 3]. The default is sticky off.
Step 7	`Router(config-slb-vserver)# sticky` *group-number* **reverse**	(Optional) Ensures that the CSM changes connections in the appropriate direction back to the same source.
Step 8	`Router(config-slb-vserver)# client` *ip-address network-mask* [**exclude**]	(Optional) Restricts which clients are allowed to use the virtual server[2 3].
Step 9	`Router(config-slb-vserver)# slb-policy` *policy-name*	(Optional) Associates one or more content switching policies with a virtual server[2].
Step 10	`Router(config-slb-vserver)# inservice`	Enables the virtual server for use by the CSM[2].
Step 11	`Router# show module csm` *slot* **vserver** [**details**]	Displays information for virtual servers defined for Content Switching.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.

2. The **no** form of this command restores the defaults.
3. These parameters refer to the default policy.

This example shows how to configure a virtual server named barnett, associate it with the server farm named bosco, and configure a sticky connection with a duration of 50 minutes to sticky group 12:

```
Router(config)# mod csm 2
Router(config-module-csm)# sticky 1 cookie foo timeout 100
Router(config-module-csm)# exit
Router(config-module-csm)#
Router(config-module-csm)# serverfarm bosco
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)#
Router(config-slb-sfarm)# vserver barnett
Router(config-slb-vserver)# virtual 10.1.0.85 tcp 80
Router(config-slb-vserver)# serverfarm bosco
Router(config-slb-vserver)# sticky 50 group 12
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
Router(config-module-csm)# end
```

This example shows how to configure a virtual server, named vs1, with two policies and a default server farm when client traffic matches a specific policy. The virtual server will be load balanced to the server farm attached to that policy. When client traffic fails to match any policy, the virtual server will be load balanced to the default server farm named bosco.

```
Router(config)# mod csm 2
Router(config-module-csm)# map map3 url
Router(config-slb-map-url)# match protocol http url *finance*
Router(config-slb-map-url)#
Router(config-slb-map-url)# map map4 url
Router(config-slb-map-url)# match protocol http url *mail*
Router(config-slb-map-url)#
Router(config-slb-map-url)# serverfarm bar1
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-real)# inservice
Router(config-slb-real)#
Router(config-slb-real)# serverfarm bar2
Router(config-slb-sfarm)# real 10.1.0.106
Router(config-slb-real)# inservice
Router(config-slb-real)#
Router(config-slb-real)# serverfarm bosco
Router(config-slb-sfarm)# real 10.1.0.107
Router(config-slb-real)# inservice
Router(config-slb-real)#
Router(config-slb-real)# policy pc1
Router(config-slb-policy)# serverfarm bar1
Router(config-slb-policy)# url-map map3
Router(config-slb-policy)# exit
Router(config-module-csm)#
Router(config-module-csm)# policy pc2
Router(config-slb-policy)# serverfarm bar2
Router(config-slb-policy)# url-map map4
Router(config-slb-policy)# exit
Router(config-module-csm)#
Router(config-module-csm)# vserver bar1
Router(config-slb-vserver)# virtual 10.1.0.86 tcp 80
Router(config-slb-vserver)# slb-policy pc1
Router(config-slb-vserver)# slb-policy pc2
```

```
Router(config-slb-vserver)# serverfarm bosco
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)#
```

Configuring TCP Parameters

Transmission Control Protocol (TCP) is a connection-oriented protocol that uses known protocol messages for activating and deactivating TCP sessions. In server load balancing, when adding or removing a connection from the connection database, the Finite State Machine correlates TCP signals such as SYN, SYN/ACK, FIN, and RST. When adding connections, these signals are used for detecting server failure and recovery and for determining the number of connections per server.

The CSM also supports User Datagram Protocol (UDP). Because UDP is not connection-oriented, protocol messages cannot be generically sniffed (without knowing details of the upper-layer protocol) to detect the beginning or end of a UDP message exchange. Detection of UDP connection termination is based on a configurable idle timer. Protocols requiring multiple simultaneous connections to the same real server are supported (such as FTP). Internet Control Management Protocol (ICMP) messages destined for the virtual IP address are also handled (such as ping).

To configure TCP parameters, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# vserver` *virtserver-name*	Identifies the virtual server and enters the virtual server configuration mode[1,2].
Step 2	`Router(config-slb-vserver)# idle` *duration*	Configures the amount of time (in seconds) that connection information is maintained in the absence of packet activity for a connection[2].

1. Enter the **exit** command to leave a mode or submode. To return to the Router (config)> top level of the menu, enter the **end** command.
2. The **no** form of this command restores the defaults.

This example shows how to configure TCP parameters for virtual servers:

```
Router(config-module-csm)# vserver barnett
Router(config-slb-vserver)# idle 10
```

Configuring Dynamic Feedback Protocol

Configuring the Dynamic Feedback Protocol (DFP) allows servers to provide feedback to the CSM to enhance load balancing. DFP allows host agents (residing on the physical server) to dynamically report the change in status of the host systems providing a virtual service.

Note A DFP agent may be on any host machine. A DFP agent is independent of the IP addresses and port numbers of the real servers that are managed by the agent. DFP Manager is responsible for establishing the connections with DFP agents and receiving load vectors from DFP agents.

To configure DFP, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# dfp [password password]`	Configures DFP manager, supplies an optional password, and enters the DFP agent submode[1, 2].
Step 2	`Router(config-slb-dfp)# agent ip-address port [activity-timeout [retry-count [retry-interval]]]`	Configures the time intervals between keepalive messages, the number of consecutive connection attempts or invalid DFP reports, and the interval between connection attempts[2].
Step 3	`Router# show module csm slot dfp [agent [detail | ip-address port] | manager [ip_addr] | detail | weights]`	Displays DFP manager and agent information.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example shows how to configure the dynamic feedback protocol:

```
Router(config-module-csm)# dfp password password
Router(config-slb-dfp)# agent 123.234.34.55 5 6 10 20
Router(config-slb-dfp)# exit
```

Configuring Redirect Virtual Servers

The **redirect-vserver** command is a server farm submode command that allows you to configure virtual servers dedicated to real servers. This mapping provides connection persistence, which maintains connections from clients to real servers across TCP sessions.

To configure redirect virtual servers, perform this task:

	Command	Purpose
Step 1	`Router(config-slb-sfarm)# redirect-vserver name`	Configures virtual servers dedicated to real servers and enters the redirect server submode[1, 2].
Step 2	`Router(config-slb-redirect-v)# webhost relocation relocation string`	Configures the destination URL host name when redirecting HTTP requests arrive at this server farm. Only the beginning of the URL can be specified in the relocation string. The remaining portion is taken from the original HTTP request[2].
Step 3	`Router(config-redirect-v)# webhost backup backup string`	Configures the relocation string sent in response to HTTP requests in the event that the redirect server is out of service. Only the beginning of the relocation string can be specified. The remaining portion is taken from the original HTTP request[2].
Step 4	`Router(config-redirect-v)# virtual v_ipaddress tcp port`	Configures the redirect virtual server IP address and port[2].
Step 5	`Router(config-redirect-v)# idle duration`	Sets the CSM connection idle timer for the redirect virtual server[2].

	Command	Purpose
Step 6	`Router(config-redirect-v)# client ip-address network-mask [exclude]`	Configures the combination of the *ip-address* and *network-mask* used to restrict which clients are allowed to access the redirect virtual server[2].
Step 7	`Router(config-redirect-v)# inservice`	Enables the redirect virtual server and begins advertisements[2].
Step 8	`Router(config-redirect-v)# ssl port`	(Optional) Enables SSL forwarding by the virtual server.
Step 9	`Router# show module csm vserver redirect [detail]`	Shows all redirect servers configured.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example shows how to configure redirect virtual servers to specify virtual servers to real servers in a server farm:

```
Router (config)# serverfarm FARM1
Router (config-slb-sfarm)# redirect-vserver REDIR_1
Router (config-slb-redirect-)# webhost relocation 127.1.2.30 301
Router (config-slb-redirect-)# virtual 172.1.2.30 tcp www
Router (config-slb-redirect-)# inservice
Router (config-slb-redirect-)# exit
Router (config-slb-sfarm)# redirect-vserver REDIR_2
Router (config-slb-redirect-)# webhost relocation 127.1.2.31 301
Router (config-slb-redirect-)# virtual 172.1.2.31 tcp www
Router (config-slb-redirect-)# inservice
Router (config-slb-redirect-)# exit
Router (config-slb-sfarm)# real 10.8.0.8
Router (config-slb-real)# redirect-vserver REDIR_1
Router (config-slb-real)# inservice
Router (config-slb-sfarm)# real 10.8.0.9
Router (config-slb-real)# redirect-vserver REDIR_2
Router (config-slb-real)# inservice
Router (config-slb-real)# end
Router# show module csm serverfarm detail
```

Configuring Client NAT Pools

When you configure client Network Address Translation (NAT) pools, NAT converts the source IP address of the client requests into an IP address on the server-side VLAN. Use the NAT pool name in the *serverfarm* submode of the **nat** command to specify which connections need to be configured for client NAT pools.

To configure client NAT pools, perform this task:

	Command	Purpose
Step 1	`Router(config-module-csm)# natpool pool-name start-ip end-ip netmask mask`	Configures a content switching NAT. You must create at least one client address pool to use this command[1, 2].
Step 2	`Router(config-module-csm)# serverfarm serverfarm-name`	Enters the *serverfarm* submode to apply the client NAT.

	Command	Purpose
Step 3	`Router(config-slb-sfarm)# nat client clientpool-name`	Associates the configured NAT pool with the server farm.
Step 4	`Router# show module csm natpool [name pool-name] [detail]`	Displays the NAT configuration.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.

This example shows how to configure client NAT pools:

```
Router(config)# natpool pool1 102.36.445.2 102.36.16.8 netmask 255.255.255.0
Router(config)# serverfarm farm1
Router(config-slb-sfarm)# nat client pool1
```

Configuring Server-Initiated Connections

NAT for the server allows you to support connections initiated by real servers and to provide a default configuration used for servers initiating connections that do not have matching entries in the server NAT configuration. By default, the CSM allows server-originated connections without NAT.

To configure NAT for the server, perform this task:

	Command	Purpose
Step 1	`Router(config)# static [drop | nat [ipaddress | virtual]]`	Configures the server-originated connections. Options include dropping the connections, configuring them with NAT with a given IP address, or with the virtual IP address that they are associated with[1, 2].
Step 2	`Router(config-slb-static)# real ip-address [subnet-mask]`	Configures the *static nat* submode where the servers will have this NAT option. You cannot use the same real server with multiple NAT configuration options.

1. Enter the **exit** command to leave a mode or submode. Enter the **end** command to return to the menu's top level.
2. The **no** form of this command restores the defaults.