Configuring the Cisco Content Services Gateway

This chapter describes how to configure the Cisco Content Services Gateway (CSG) and contains the following sections:

- Preparing to Configure the CSG, page 3-1
- Upgrading to a New CSG Release, page 3-3
- Saving and Restoring Configurations, page 3-6
- Configuring the CSG, page 3-6
- Protocol-Specific Configuration Details, page 3-28
- Other Configuration Tasks, page 3-35
- Configuration Examples, page 3-39

Preparing to Configure the CSG

Before you configure the CSG, perform the following actions:

- Make sure that the Cisco IOS version for the switch matches that of the module. You must use Cisco IOS Release 12.1(12c)E4 or later.
- Configure VLANs on the Catalyst 6000 series switch or Cisco 7600 series router *before* you configure VLANs for the CSG. VLAN IDs must be the same for the switch and the module. For details, see the *Catalyst 6000 Series IOS Software Configuration Guide* or the *Cisco 7600 Series Cisco IOS Software Configuration Guide*.

  The following example shows how to configure VLANs:

```
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
```

- Assign physical interfaces that connect to the servers and to the clients to the corresponding VLAN.

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

```
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-side VLAN or the server-side VLAN, then you must configure the corresponding Layer 3 VLAN interface.

Caution If you use the MSFC as the router for both the client side and the server side at the same time, you must ensure that packets for billable flows cannot bypass the CSG. Also, if you use static **ip route** commands to switch traffic to the CSGs, packets might loop between the MSFC and the CSG in this configuration. To avoid these problems, use other routing techniques to switch packets to the CSG, such as policy-based routing.

The following example shows how to configure the Layer 3 VLAN interface:

```
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 CSG is the Cisco IOS command-line interface (CLI). For more information about using the CLI and Cisco IOS command modes, see Chapter 2 in the *Catalyst 6000 Series IOS Software Configuration Guide*, and Chapter 2 in the *Cisco 7600 Series Cisco IOS Software Configuration Guide.*

Accessing Online Help

In any command mode, you can enter the question mark (?) at the prompt to see a list of available commands. For example:

```
Router> ?
```

or

```
Router(config)# ip csg ?
```

The online help shows the default configuration values and the ranges that are available for the commands.

Upgrading to a New CSG Release

This section describes the following methods for upgrading the CSG:

- Upgrading from the Supervisor Engine Boot Flash Memory, page 3-3
- Upgrading from a Flash PC Memory Card, page 3-4
- Upgrading from an External TFTP Server, page 3-5
- Upgrading from CSG 3.1(3)C6(2) to the CSG 3.1(3)C7(1), page 3-5
- Performing a Hitless Upgrade, page 3-6

During the upgrade, enter all commands on a console that is connected to the Supervisor Engine. Enter each configuration command on a separate line.

Note To complete the upgrade, enter the **exit** command to return to the Supervisor Engine prompt. If you do not terminate the session, and you remove the CSG from the Catalyst 6000 series chassis, you cannot enter configuration commands to the CSG unless you press **Ctrl-Shift-6**, enter **x**, and enter the **disconnect** command at the prompt.

The CSG can run in hybrid mode, with CatOS on the Supervisor Engine and Cisco IOS software on the MSFC. In the CSG hybrid mode, you can upgrade the CSG only from the MSFC. To enter the MSFC console from CatOS, enter **switch console**. After you enter the MSFC console, you can configure the CSG as you would in native mode. To exit the MSFC console, enter **Ctrl-c** three times.

For redundant MSFC configurations, you cannot upgrade older versions of the CSG from the MSFC in slot 2 by using the keyword **slot0:**. You can either upgrade from the MSFC in slot 1, or you can upgrade using IP address 127.0.0.22.

Upgrading from the Supervisor Engine Boot Flash Memory

For instructions on loading images into boot flash memory, see the *Catalyst 6000 Family Flash Card Install Note* or the *Cisco 7600 Series Cisco IOS Software Configuration Guide*.

To upgrade the CSG from the Supervisor Engine boot flash memory, perform these steps:

Step 1 Enable the TFTP server to supply the image from the boot flash memory:

```
Router> enable
Router# configure terminal
Router(config)# tftp-server bootflash:name
```

where *name* is the CSG image name, such as c6csg-apc.31-3.c7.1.

Step 2 Establish a session between the Supervisor Engine and the CSG:

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

where *slot-number* is the slot number for the CSG that you want to upgrade.

Step 3 Load the image from the Supervisor Engine to the CSG:

```
CSM# upgrade 127.0.0.yz name
```

where:

- *y* is the slot number—**1** for slot 1, **2** for slot 2, and so on.
- *z* identifies the Supervisor Engine—**1** for Supervisor Engine 720, **2** for Supervisor Engine 32.
- *name* is the CSG image name, such as c6csg-apc.31-3.c7.1.

Step 4 Reboot the CSG by turning it off then turning it back on, or reboot it by entering the following command on the Supervisor Engine console:

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

where *slot-number* is the slot number for the CSG that has been upgraded.

Upgrading from a Flash PC Memory Card

To upgrade the CSG from a removable flash PC memory card inserted in the Supervisor Engine, perform these steps:

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

```
Router> enable
Router# configure terminal
Router(config)# tftp-server slotx:name
```

where:

- *x* is the slot number for the CSG2 that you want to upgrade.
- *name* is the CSG2 image name, such as c6csg-apc.31-3.c7.1.

Step 2 Establish a session between the Supervisor Engine and the CSG:

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

where *slot-number* is the slot number for the CSG that you want to upgrade.

Step 3 Load the image from the Supervisor Engine to the CSG:

```
CSM# upgrade 127.0.0.yz name
```

where:

- *y* is the slot number—**1** for slot 1, **2** for slot 2, and so on.
- *z* identifies the Supervisor Engine—**1** for Supervisor Engine 720, **2** for Supervisor Engine 32.
- *name* is the CSG image name, such as c6csg-apc.31-3.c7.1.

Step 4 Reboot the CSG by turning it off then turning it back on, or reboot it by entering the following command on the Supervisor Engine console:

```
Router# configure terminal
Router# hw-module module slot-number reset
```

where *slot-number* is the slot number for the CSG that has been upgraded.

Upgrading from an External TFTP Server

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

Step 1 Create a VLAN on the Supervisor Engine for the TFTP CSG runtime image download.

Note You can use an existing VLAN. However, for a reliable download, we recommend that you 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 CSG **vlan** command. See the "Configuring VLANs" section on page 3-36 for more information.

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

Step 6 (Optional) Add a route to the TFTP server for the CSG, if necessary.

Step 7 Enter the **show csg** *slot* **vlan detail** command to verify your configuration. See the "Configuring VLANs" section on page 3-36 for more information.

Step 8 Establish a Telnet connection to the CSG by using the **session** *slot-number* **0** command.

Step 9 Upgrade the image by using the **upgrade** *TFTP-server-IP-address* **c6csg-apc.***revision***.bin** command, where *revision* is **31-3.c7.1** if you are using the CSG 3.1(3)C7(1).

Step 10 Reboot the CSG.

For the CSG hybrid mode, you must enable the VLAN for the CSG from the CatOS console by entering the following command:

set vlan *vlan-list*

To add a VLAN, enter the following command:

set trunk *slot*/**1** *vlan-list*

To reset a VLAN, enter the following command:

clear trunk *slot*/**1** *vlan-list*

Upgrading from CSG 3.1(3)C6(2) to the CSG 3.1(3)C7(1)

The CSG 3.1(3)C7(1) requires one of the following supervisor engines, and a module with ports to connect server and client networks, running Cisco IOS Release 12.2(18)SXF1 or Cisco IOS Release 12.2(18)SRA:

- A Supervisor Engine 32 with an MSFC2A and PFC3B (WS-SUP32-GE-3B/MSFC2A/PFC3B or WS-SUP32-10GE-3B/MSFC2A/PFC3B)
- A Supervisor Engine 720 with an MSFC3-BXL (SUP720-MSFC3-BXL)

The CSG 3.1(3)C7(1) does not support Cisco IOS Releases 12.2(17d)SXB, 12.2(17d)SXB1, 12.2(18)SXD, and 12.2(18)SXE. Therefore, you must upgrade to a Supervisor Engine running Cisco IOS Release 12.2(18)SXF1 or Cisco IOS Release 12.2(18)SRA, either before or at the same time that you upgrade the CSG.

Even if you keep your existing configuration and you do not enable any new CSG 3.1(3)C7(1) features, you must be aware of the following differences between the CSG 3.1(3)C7(1) and CSG 3.1(3)C6(2):

- All new CSG 3.1(3)C7(1) TLVs are optional. They cause no backward compatibility issues with entities that support previous releases of the interface, provided those entities ignore unrecognized TLVs and messages.

Performing a Hitless Upgrade

A *hitless upgrade* allows you to upgrade to a new version without major service disruption resulting from the downtime for the upgrade. To perform a hitless upgrade, perform these steps:

Step 1 Perform a write memory on standby.

Step 2 Upgrade the standby system with the new release, and then reboot the CSG. The standby CSG boots as standby with the new release.

Step 3 After rebooting, wait for all of the information to propagate to the standby. Be aware that it might take up to an hour for this process to complete.

Step 4 Upgrade the active CSG with the new release, and then reboot the active CSG. When the active CSG reboots, the standby CSG becomes the new active CSG and takes over the service responsibility.

The rebooted CSG comes up as standby.

Saving and Restoring Configurations

For information about saving and restoring configurations, see the *Catalyst 6000 Series IOS Software Configuration Guide* or the *Cisco 7600 Series Cisco IOS Software Configuration Guide*.

Configuring the CSG

You must perform the following tasks before you can use the content billing feature on the CSG:

- Specifying CSG Locations, page 3-7
- Configuring User Groups, page 3-7
- Configuring Accounting Policies, page 3-10
- Activating the Accounting Policy on the CSG, page 3-11
- Defining Client and Server Connectivity, page 3-12
- Downloading an Accounting Service, page 3-12
- Downloading Ruleset Content, page 3-13
- Configuring Policies and Traffic Types, page 3-13

- Configuring a Content Billing Service, page 3-14
- Configuring Content, page 3-15
- Configuring Fixed or Variable Format CDR Support, page 3-16
- Configuring a Refund Policy on the CSG, page 3-17
- Configuring RADIUS Accounting Attribute and VSA Subattribute Reporting, page 3-18
- Configuring RADIUS Proxy, page 3-19
- Configuring RADIUS Endpoint, page 3-20
- Configuring HTTP Header Reporting, page 3-20
- Configuring a Ruleset, page 3-20
- Configuring Maps for Pattern-Matching, page 3-21
- Configuring a Symbolic Weight Name, page 3-24
- Configuring Advice of Charge, Filtering, and Other Per-Event Authorizations, page 3-25
- Configuring Quota Server Load Sharing, page 3-26
- Configuring Service-Level CDR Summarization, page 3-26
- Configuring Quota Server Reauthorization, page 3-27

Specifying CSG Locations

Before you can enter CSG configuration commands on the switch, you must specify the CSG that you want to configure.

To specify the slot number of a CSG, follow these steps:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# module csg` *slot-number*	Enters module CSG configuration mode for a specified slot. All further configuration commands that you enter after this point apply to the CSG installed in the slot that you specified with *slot-number*.

Note Unless otherwise specified, all the examples in this guide assume that you have already entered this command and have entered the configuration mode for the CSG that you are configuring.

Configuring User Groups

To configure the CSG to record and generate accounting records, you must specify the user groups for which you want to generate accounting records, as well as the user database that you want the CSG to query for user IDs.

To configure user groups on the CSG; to specify the user database, RADIUS endpoint, and quota servers; and to configure redirection of NAT flows, perform the following steps:

	Command	Purpose
Step 1	`Router(config)# ip csg user-group group-name`	Defines a CSG user group and specifies a user database name.
Step 2	`Router(config-csg-group)# database ip-address port-number`	Specifies the location of the user database, including the IP address and port number of the user database.
Step 3	`Router(config-csg-group)# entries {idle duration [pod] | max entries-number}`	(Optional) Defines settings for the CSG User Table.
Step 4	`Router(config-csg-group)# quota local-port port-number`	(Optional) Configures the local port on which the CSG receives communications from quota servers.
Step 5	`Router(config-csg-group)# quota server {ip-address port-number priority | reassign}`	(Optional) Configures the quota servers that return billing quota values for users. **Note** The CSG does not support multiple quota servers with the same IP address.
Step 6	`Router(config-csg-group)# quota activate number`	(Optional) Allows load balancing of quota servers, similar to the BMA load-balancing feature. Multiple quota servers can be simultaneously active, and the CSG assigns a quota server to each user. All quota transactions for the user are handled by the same quota server. When a quota server fails, the transactions of all users associated with that quota server are distributed among other quota servers. Range for the *number* argument is 1 through 10. **Note** Multiple quota servers cannot have the same IP address.
Step 7	`Router(config-csg-group)# radius acct-port port-number`	Specifies the port number for the RADIUS Accounting endpoint.
Step 8	`Router(config-csg-group)# radius handoff [duration]`	(Optional) Configures RADIUS handoff support.
Step 9	`Router(config-csg-group)# radius key secret`	Configures the CSG to be the RADIUS endpoint for accounting records, and provides the key.
Step 10	`Router(config-csg-group)# radius parse strict`	(Optional) Tightens the parsing rules for RADIUS flows.
Step 11	`Router(config-csg-group)# radius pod attribute radius-attribute-number`	(Optional) Specifies the RADIUS attributes to be copied from the RADIUS Start message and sent to the NAS in the Packet of Disconnect (PoD).
Step 12	`Router(config-csg-group)# radius pod nas [start-ip end-ip] port key [encrypt] secret-string`	(Optional) Specifies the NAS port to which the CSG is to send the Packet of Disconnect (PoD) message, and the key to use in calculating the Authenticator.
Step 13	`Router(config-csg-group)# radius pod timeout timeout retransmit retransmit`	(Optional) Specifies the number of times to retry the RADIUS Packet of Disconnect (PoD) message if it is not acknowledged by means of an ACK message, and the interval between retransmissions.
Step 14	`Router(config-csg-group)# radius server ip-address [port-number]`	(Optional) Enables RADIUS proxy.
Step 15	`Router(config-csg-group)# radius userid {1 | 31 | User-Name | Calling-Station-Id}`	(Optional) RADIUS attribute used to extract the user IDs from a RADIUS record.

	Command	Purpose
Step 16	`Router(config-csg-group)# radius start restart session-id {attr-number | {26 | vsa} {vendor-id | 3gpp} subattr-number}`	(Optional) Deletes an existing User Table entry for a specific user (when a RADIUS Accounting Start is received), and creates a new entry for that user.
Step 17	`Router(config-csg-group)# radius stop purge {attr-number | {26 | vsa} {vendor-id | 3gpp} subattr-number}`	(Optional) Specifies the attribute (which might be a vendor-specific attribute [VSA]) that must be included in the RADIUS Accounting Stop request in order for the User Table entry to be deleted.
Step 18	`Router(config-csg-group)# radius monitor server-addr server-port [key [encrypt] secret-string]`	Specifies that the CSG is to monitor the RADIUS flows to the specified server.
Step 19	`Router(config-csg-group)# redirect nat ip-address [port-number]`	(Optional) Redirects client NAT flows to an alternate IP address when the client's quota is exhausted.
Step 20	`Router(config-csg-group)# redirect http url`	(Optional) Redirects client HTTP flows to an alternate URL when the client's quota is exhausted.
Step 21	`Router(config-csg-group)# redirect wap url`	(Optional) Redirects client WAP flows to an alternate URL when the client's quota is exhausted.
Step 22	`Router(config-csg-group)# aoc confirmation`	Configures a token for use in Advice of Charge (AoC) URL-rewriting.
Step 23	`Router(config-csg-group)# user-profile server {quota | radius {remove | pass}}`	(Optional) Specifies which server is used to obtain the user profile or billing plan. **Note** The VSA is removed from the Access-Accept message only if **remove** is specified. We recommend that you use **pass** to reduce processing time on the CSG. Use **remove** only if the RADIUS client rejects the Cisco VSA in the message. Additionally, the user ID must be in the message that contains the billing plan.

The following example shows how to configure a CSG user group, including a database, a RADIUS endpoint, quota servers, and NAT redirect:

```
ip csg user-group G1
 entries max 100000
 database 10.1.2.3 11111
 quota local-port 6666
 quota server 10.1.4.5 888 1
 quota server 10.1.6.7 999 2
 radius acct-port 7777
 radius key SECRET_PASSWORD
 radius parse strict
 radius server 10.13.14.15
 radius userid User-Name
 redirect nat 10.33.33.3
!
```

```
ip csg user-group U1
 radius userid User-Name
 radius monitor 10.2.3.4 1234 key cisco
 radius monitor 10.2.3.9 1234 key cisco2
 radius monitor 10.2.7.4 3901 key cisco
```

Configuring Accounting Policies

To configure the CSG to record and generate accounting records, you must configure content-based client accounting as a service. You must specify the user groups for which you want to generate accounting records, as well as the Billing Mediation Agent (BMA) to which you want the accounting records sent.

To configure the accounting policies on the CSG, perform the following steps:

	Command	Purpose
Step 1	`Router(config)# ip csg accounting` *name*	Configures content-based client accounting as a policy.
Step 2	`Router(config-csg-accounting)# user-group` *name*	Associates a user group with a specific accounting service.
Step 3	`Router(config-csg-accounting)# agent` *ip-address port-number priority*	Defines the active and standby Billing Mediation Agents (BMAs) to which billing records are to be sent. **Note** The CSG does not support multiple agents with the same IP address.
Step 4	`Router(config-csg-accounting)# agent activate` [*number* [**sticky** *seconds*]]	(Optional) Enables support for multiple active BMAs.
Step 5	`Router(config-csg-accounting)# agent local-port` *port-number*	(Optional) Defines the port on which the CSG will listen for packets from the BMAs.
Step 6	`Router(config-csg-accounting)# keepalive` *number-of-seconds*	(Optional) Defines the keepalive time interval (in seconds) that will be used to test the health of BMAs.
Step 7	`Router(config-csg-accounting)# records batch`	(Optional) Batches billing records into a single message before sending them to the BMA.
Step 8	`Router(config-csg-accounting)# records http-statistics`	(Optional) Sends the HTTP Statistics data record to the BMA.
Step 9	`Router(config-csg-accounting)# records intermediate` {**bytes** *bytes* | **time** *seconds* | **bytes** *bytes* **time** *seconds*}	(Optional) Enables the generation of intermediate billing records.
Step 10	`Router(config-csg-accounting)# records max` *number*	(Optional) Defines the maximum number of billing records that can be stored or queued in the CSG before they are forwarded to the Billing Mediation Agent (BMA). If the number of queued records exceeds the *number* argument, the CSG tries to forward the records to the Persistent Storage Device (PSD), if one is available. Otherwise, the CSG discards the billing records.
Step 11	`Router(config-csg-accounting)# records format`	(Optional) Specifies variable, fixed, or variable-single CDR format.
Step 12	`Router(config-csg-accounting)# record-storage` *ip-address* [*port*]	(Optional) Defines a Cisco Persistent Storage Device (PSD) to associate with this accounting group.

	Command	Purpose
Step 13	`Router(config-csg-accounting)# record-storage local-port port`	(Optional) Defines the source port that the CSG will use for communicating with the record store.
Step 14	`Router(config-csg-accounting)# report http header header-name`	(Optional) Defines the inclusion of multiple HTTP request headers in the CSG HTTP_Header CDR.
Step 15	`Router(config-csg-accounting)# report radius attribute radius-attribute-number`	(Optional) Specifies the RADIUS attributes to be copied from the RADIUS Start message and sent to the BMA in each billing record.
Step 16	`Router(config-csg-accounting)# report usage {bytes ip | seconds}`	(Optional) Enables supplemental usage reporting.
Step 17	`Router(config-csg-accounting)# inservice`	Activates the accounting service on a CSG.

Step 18	`Router# show module csg slot accounting {agent | database | error | quota-server | radius | users {all | statistics | ip-address [ipmask] | userid userid}} [detail] [module num]`
	
	or
	
	`Router# show ip csg accounting {agent | database | error | quota-server | radius | users {all | statistics | ip-address [ipmask] | userid userid}} [detail] [module num]`
		Displays information for the CSG billing feature.

The following example shows how to define the CSG accounting policy:

```
ip csg accounting A1
 user-group G1
 agent activate 2
 agent local-port 3775
 agent 10.1.2.4 11112 1
 agent 10.1.2.5 11113 2
 keepalive 3
 records batch
 records http-statistics
 records intermediate bytes 100000 time 3600
 records max 250
 record-storage local-port 5002
 record-storage 172.18.12.226
 report http header x-subno
 report http header x-al-session-id
 report radius attribute 3
 report radius attribute 5
 inservice
```

Activating the Accounting Policy on the CSG

To activate the accounting policy on the CSG, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# accounting service-name`	Downloads a configured accounting service to a CSG card.

Defining Client and Server Connectivity

To properly configure the CSG, you must create VLANs for both the client side and the server side of the switch. You must do this so that the CSG knows where to forward the traffic it receives. The minimal configuration requires one client-side VLAN and one server-side VLAN. You must also configure IP addresses for the VLANs and for all gateways.

To configure client-side VLANs on the CSG, enter the following commands in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# vlan vlan-id client [vlan-name]`	Configures the client-side VLANs and enters the client VLAN mode. **Note** You cannot use VLAN 1 as a client-side VLAN for the CSG.

Then configure an IP address on this client-side VLAN.

To configure server-side VLANs on the CSG, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# vlan vlan-id server [vlan-name]`	Configures the server-side VLANs and enters the server VLAN mode. **Note** You cannot use VLAN 1 as a server-side VLAN for the CSG.

Then configure an IP address on this server-side VLAN.

The following example shows how to configure client VLANs and server VLANs:

```
vlan 10 server
 ip address 10.250.0.1 255.255.0.0
 gateway 10.250.1.1

vlan 251 client
 ip address 10.251.0.1 255.255.0.0
 route 10.200.0.0 255.254.0.0 gateway 10.251.2.11
```

Downloading an Accounting Service

Before you can configure the CSG to perform content billing, you must enable it to reference and download a specific accounting service configuration.

To install the accounting service in a specific CSG, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# accounting service-name`	Assigns a specific accounting service to a specific CSG.

Downloading Ruleset Content

A CSG billing *ruleset* is a list of all the content names that are to be downloaded to a specific CSG card.

To download all ruleset-defined content to a CSG card, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# ruleset ruleset-name`	Downloads all content configured by a ruleset to a CSG card.

Configuring Policies and Traffic Types

Policies are access rules that traffic must match in order to be handled by a specific server farm. Policies allow the CSG to apply filters to certain types of traffic subject to the accounting service.

When the CSG matches 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 configured in the content. You can reorder the policies in the list by removing policies and reentering them in the order that you prefer.

To configure accounting records policies, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg policy policy-name`	Defines a policy for qualifying flows for CSG accounting services, and enters CSG policy configuration mode.
Step 2	`Router(config-csg-policy)# accounting [type {http | ftp | wap {connection-oriented | connectionless} | rtsp | ftp | smtp | pop3 | other}] [customer-string string]`	Defines the accounting type and a customer string for all flows that comply with a CSG billing policy.
Step 3	`Router(config-csg-policy)# client-group {std-access-list-number | std-access-list-name}`	References a standard access list that is part of a CSG billing policy.
Step 4	`Router(config-csg-policy)# client-ip http-header x-forwarded-for`	Specifies that the user's IP address is to be obtained from the URL header after the **x-forwarded-for** keyword.
Step 5	`Router(config-csg-policy)# header-map header-map-name`	References a header map that is part of a CSG billing policy.
Step 6	`Router(config-csg-policy)# next-hop ip-address`	Defines a next-hop IP address.
Step 7	`Router(config-csg-policy)# url-map url-map-name`	References a URL map that is part of a CSG billing policy.

The following example shows how to define a policy:

```
ip csg policy MOVIES_COMEDY
 accounting type http customer-string MOVIES_COMEDY
 client-group 44
 client-ip http-header x-forwarded-for
 header-map MOVIES
 next-hop 33.0.0.150
 url-map MOVIES
```

Configuring a Content Billing Service

A CSG content billing service is a component of a billing plan to which users subscribe.

You can configure one or more content billing services for the CSG. Each service represents a group of content that is billed the same way, such as billing per-click (or per-request) or billing per-IP byte, and that shares part of a user's quota. Grouping content into one or more services enables you to separate, for example, a user's prepaid quota for Internet browsing from his quota for e-mails.

For each service, the CSG downloads a separate quota, and deducts from that quota. Quotas are specified in units called *quadrans*. A quadran is a generic unit whose "value" is defined by each quota server. A quadran can represent, for example, a click for a per-click service (for example, an HTTP request), or a byte for a per-volume service. The value of a quadran is transparent to the CSG; the CSG simply requests and downloads quadrans as needed from quota servers.

The CSG requests an additional quota grant when a user's per-click quota falls below a specified percentage of the last quota grant, or when a user's per-volume quota falls below a specified percentage of the last quota grant or 32 KB, whichever is greater.

For each service that a user tries to access, the CSG maintains a separate logical accounting session. When a user's quota is divided among multiple services, the CSG requests an additional quota grant for each service individually, based on its usage.

If a user fails authorization for a service, but continues to send new requests for that service, the CSG waits a specified time before sending the quota server a reauthorization request for that user. This ensures that the quota server is not inundated with reauthorization requests from unauthorized users.

The billing basis specifies how billing is to be charged:

- Per-click (fixed-cost) billing is charged at a fixed cost, which is deducted each time the first packet for a transaction hits a content-policy pair (that is, deducted for each request).
- Volume-based billing can be based on either the number of IP bytes or the number of TCP bytes.
- Duration-based billing can be based on either service duration time or connection duration time.
- The **exclude mms** option specifies that Multimedia Messaging Service (MMS) content over wireless application protocol (WAP) is not billed.

To configure a content billing service, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg service service-name`	Configures a content billing service, and enters CSG service configuration mode.
Step 2	`Router(config-csg-service)# content content-name policy policy-name [weight weight-name]`	Configures content as a member of a CSG billing service, identifies a policy to apply to this content, and optionally assigns a weight to this content.
Step 3	`Router(config-csg-service)# basis {byte {ip | tcp} | {fixed | second [connect] [exclude mms]}`	(Optional) Specifies the billing basis for a CSG content billing service. **Note** When changing the basis for a service, the content must be taken out of service.
Step 4	`Router(config-csg-service)# idle duration`	(Optional) Specifies the minimum amount of time that the CSG maintains a service with no user sessions.

The CSG allows you to define a pool of up to 1024 services. You can authorize each user for any number of services from that pool, but we recommend that the billing system not authorize each user for more than 10 active services. Exceeding this guideline could lead to the following problems:

- The increase in the number of quota authorizations per user can overload both the quota server and the CSG.
- As the number of services for which a user is actively authorized increases, the user's quota becomes fragmented. Although the CSG allows the billing system to recall and redistribute the quota, so that the user is not denied service because of quota fragmentation, the process increases overhead in both the quota server and the CSG.

The following example shows how to configure a content billing service:

```
ip csg service MOVIES
 basis fixed
 content MOVIES_COMEDY policy MOVIES_COMEDY
 content MOVIES_ACTION policy MOVIES_ACTION weight DOUBLE
 idle 120
```

Configuring Content

The CSG uses the Cisco command-line interface (CLI), and requires content configurations or virtual server configurations. This section provides information about configuring content.

A CSG content configuration contains the following information:

- Layer 3 information that specifies the IP-level details of the content.
- Layer 4 information that specifies transport layer parameters, such as TCP and User Datagram Protocol (UDP) port numbers.

If the content configuration does not match any service listed under a user's billing plan, the CSG considers the service to be either free or postpaid, and the CSG does not try to authorize the user with the quota server.

To configure content for a CSG accounting service, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg content` *content-name*	Configures content for CSG accounting services, and enters CSG content configuration mode.
Step 2	`Router(config-csg-content)# policy` *policy-name*	References a CSG billing policy.
Step 3	`Router(config-csg-content)# ip` {**any** | *ip-address* [*netmask*]} [*protocol* [*port-number last-port-number*]]	Defines the subset of Layer 3 and Layer 4 flows that can be processed by the CSG accounting services. You can define *port-number* as a single value or as a range of numbers.
Step 4	`Router(config-csg-content)# client` [**include** | **exclude**] {**any** | *ip-address* [*netmask*]}	(Optional) Defines the client IP address spaces that can use the CSG content server.
Step 5	`Router(config-csg-content)# idle` *duration*	(Optional) Specifies the minimum amount of time that the CSG maintains an idle content connection.
Step 6	`Router(config-csg-content)# pending` *timeout*	(Optional) Sets the pending connection timeout.

	Command	Purpose
Step 7	`Router(config-csg-content)# replicate connection tcp`	(Optional) Replicates the connection state for all TCP connections to the CSG content servers on the standby system.
Step 8	`Router(config-csg-content)# vlan vlan-name`	(Optional) Restricts CSG billing content to a single source VLAN.
Step 9	`Router(config-csg-content)# inservice`	Activates the content service on each CSG.
Step 10	`Router # show module csg slot content [name content-name] [detail]`	Displays statistics and counters for CSG content.

The following example shows how to configure content for a CSG accounting service:

```
ip csg content MOVIES_COMEDY
 policy POLICY1
 client 10.4.4.0 255.255.255.0
 idle 120
 ip 172.18.45.0/24 tcp 8080
 pending 300
 replicate connection tcp
 vlan MOVIES_COMEDY
 inservice
```

The following example shows how to define a range of port numbers:

```
ip csg content MULTI_PORT
 policy WAP_SRV_POLICY
 ip any udp 30000 30150
 inservice
```

Configuring Fixed or Variable Format CDR Support

The CSG supports both variable and fixed format call detail record (CDR) generation, including a fixed variable format for WAP CDRs. The same variables are reported in each CDR regardless of Wireless Session Protocol (WSP) Protocol Data Unit (PDU) type. CDRs contain zero-length variables when there is no information to report, but the same set of variables are always reported in the same sequence. To configure a specific format, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg accounting records format [variable | fixed | variable-single-cdr]`	Specifies variable, fixed, or variable-single CDR format.
Step 2	`Router(config)# module csg 3 hostname MYHOST`	Specifies a variable hostname for a CSG module.

Step 3	`Router(config)# ip csg billing FOO`
	`Router(config-csg-billing)# mode postpaid`
	`Router(config-csg-billing)# service X`
	`Router(config-csg-billing)# service Y`
		Specifies that a billing plan is postpaid or prepaid.

	Command	Purpose

Step 4	`Router(config)# ip csg service FOO`
	`Router(config-csg-service)# owner name ABC_CORP`
	`Router(config-csg-service)# owner id ABC123456`
		Specifies the owner that is responsible for the content associated with a service.
		The administrator who configures owner identification is responsible for its accuracy. Correct configuration requires that contents for this service, their policies, and any associated URL or header maps, identify all data transfers with this owner, and only data transfers with this owner.

Step 5	`Router(config)# ip csg service FOO` `Router(config-csg-service)# class 7`	Specifies a service class value.

Step 6	`Router(config)# ip csg transport-type`
	`Router(config-csg-transport-type)# assign 1.2.3.4 6`
	`Router(config-csg-transport-type)# assign 2.5.3.1 7`
	`Router(config-csg-transport-type)# assign 6.6.7.5 0`
		Classifies data traffic based on its access path by using the NAS-IP reported in RADIUS. Use the **assign** command to associate IP addresses with transport-type values. Transport-type information is reported in fixed record format CDRs.

Configuring a Refund Policy on the CSG

The prepaid error reimbursement feature allows the CSG to automatically refund quota for failed transactions, as defined by the CLI. The CSG checks them in the following order: TCP/WAP flags, Application Return Code. The CSG supports flag-based refunding for all protocols. The CSG supports return code-based refunding for all protocols except RTSP.

Note If refund is enabled for a CSG prepaid service, you cannot download more than 0x6FFFFFFF bytes of data in a given transaction.

To configure a refund policy on the CSG, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg refund`	Specifies a refund policy to apply to the various services, and enters CSG refund configuration mode.

Step 2	`Router(config)# ip csg refund COMPANY-REFUND`
	`Router(config-csg-refund)# retcode http 500 509`
	`Router(config-csg-refund)# retcode wap 0x44 0x50`
	`Router(config-csg-refund)# retcode ftp 454`
		Specifies the range of application return codes for which the CSG refunds quota for Prepaid Error Reimbursement.

Step 3	`Router(config)# ip csg refund COMPANY-REFUND`
	`Router(config-csg-refund)# flags tcp 43 00`
	`Router(config-csg-refund)# flags tcp 63 01`
	`Router(config-csg-refund)# flags tcp 80 80`
	`Router(config-csg-refund)# flags ip 80 80`
	`Router(config-csg-refund)# flags wap 0 8`
		Specifies IP, TCP, or wireless application protocol (WAP) flag bit masks and values for CSG Prepaid Error Reimbursement.

The following example shows how to configure a refund policy on the CSG:

```
ip csg refund COMPANY-REFUND
 retcode http 500 509
 retcode wap 0x44 0x50
 retcode ftp 454
 flags tcp FF 14
 flags wap FF 08
```

To enable and specify the refunding policy for a CSG prepaid service, enter the following command in CSG service configuration mode:

Command	Purpose
`Router(config-csg-service)# refund-policy` *policy-name*	Enables and specifies the refunding policy for a CSG prepaid service.

The following example shows how to configure the **refund-policy** command:

```
ip csg service BILLPERCLICK
 basis fixed
 refund-policy COMPANY-REFUND
 content ADVERTISEMENTS policy ADVERTISEMENTS weight PAYBACK
 content BOOKS policy BOOKSALES
 content BOOKS policy BOOKFREE weight FREE
 content CORPORATE policy CORPORATE weight FREE
!
ip csg service BILLBYVOLUME
 basis byte tcp
 refund-policy COMPANY-REFUND
 content BILLBYVOLUME policy BILLBYVOLUME
!
ip csg service BILLBYIPVOLUME
 basis byte
 refund-policy COMPANY-REFUND
 content INTERNET policy INTERNET
```

Configuring RADIUS Accounting Attribute and VSA Subattribute Reporting

The CSG allows you to configure a list of RADIUS Accounting attributes and VSA subattributes to be reported to the BMA and quota server in every CDR. To configure these attributes and subattributes, enter the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip csg accounting` *name*	Configures content-based client accounting as a service, and enters CSG accounting configuration mode.
Step 2	`Router(config-csg-accounting)# report radius attribute` {*radius-attribute-number* | {**26** | **vsa**} {*vendor-id* | **3gpp**} *radius-subattribute-number*}	Specifies the RADIUS attributes and VSA subattributes to be copied from the RADIUS Start message and sent to the BMA in each billing record.

The attributes are copied from the RADIUS Accounting message and are sent in each billing message to the BMA and quota server.

If the list of configured attributes changes, only new RADIUS requests are subject to the new attributes. Attributes already saved for a user continue to be reported.

When a RADIUS Start request is received, any attributes received from a previous start request are deleted. If there are multiple instances of an attribute, they are all reported. Attributes are reported in the order in which they appear in the RADIUS message.

The following example shows how to define multiple RADIUS attributes:

```
Router(config)# ip csg accounting a1
Router(config-csg-accounting)# report radius attribute 3
Router(config-csg-accounting)# report radius attribute 5
Router(config-csg-accounting)# report radius attribute 7
Router(config-csg-accounting)# report radius attribute 44
```

To specify the RADIUS attributes and VSA subattributes to be copied from the RADIUS Start message and sent to the Network Access Server (NAS) in the PoD message, enter the following command in CSG user group configuration mode:

Command	Purpose
`Router(config-csg-group)# radius pod attribute {`*radius-attribute-number* `| {26 | vsa} {`*vendor-id* `| 3gpp} `*radius-subattribute-number*`}`	Specifies the RADIUS attributes and VSA subattributes to be copied from the RADIUS Start message and sent to the Network Access Server (NAS) in the Packet of Disconnect (PoD) message. If RADIUS attribute 26 is included in the message, only the configured subattributes are saved, not the entire attribute. Therefore, only the subattributes specified using the **radius pod attribut**e command are reported or included in the PoD messages.

Configuring RADIUS Proxy

The RADIUS proxy feature lets you specify that the CSG is a proxy for RADIUS messages. To configure the RADIUS proxy feature, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# radius proxy` *csg-addr server addr* `[`*csg-source-addr*`] [key [`*encrypt*`] `*secret-string*`] [table `*table-name*`]`	Specifies that the CSG is a proxy for RADIUS messages.

Note If you specify the **user-profile server radius remove** command, you might also need to configure a key.

Configuring RADIUS Endpoint

To configure the CSG as a RADIUS Accounting endpoint, enter the following command in module CSG configuration mode:

Command	Purpose
`Router(config-csg-module)# radius endpoint` *csg-addr* **key** [*encrypt*] *secret-string* [**table** *table-name*]	Identifies the CSG as an endpoint for RADIUS Accounting messages.

Configuring HTTP Header Reporting

The CSG allows you to include multiple HTTP request headers in the CSG HTTP_Header CDR. To define HTTP reporting on the CSG, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg accounting` *name*	Configures content-based client accounting as a service, and enters CSG accounting configuration mode.
Step 2	`Router(config-csg-accounting)# report http header` *x-header*	Defines the inclusion of multiple HTTP request headers in the CSG HTTP_Header CDR. You can specify any number of headers up to 256; header names cannot exceed 224 characters.

The following example shows how to enable HTTP header reporting for virtual server VS1:

```
ip csg accounting a1
 report http header x-subno
 report http header x-al-session-id
```

Configuring a Ruleset

A CSG billing ruleset is a list of all the content names that will be downloaded to a specific CSG card.

To define a ruleset for CSG billing, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg ruleset` *ruleset-name*	Configures a CSG billing ruleset, and enters CSG ruleset configuration mode.
Step 2	`Router(config-csg-ruleset)# content` *content-name*	Adds a content reference to a CSG ruleset.

If you have defined more than one content name by using multiple **ip csg content** commands, you can also configure more than one **content** command in CSG ruleset configuration mode. The following example shows how to define a CSG billing ruleset:

```
ip csg ruleset R1
 content MOVIES_COMEDY
 content MOVIES_ACTION
```

Configuring Maps for Pattern-Matching

The CSG maps are used to match URLs or headers against a pattern, to determine whether flows will be processed by the CSG accounting services.

To define the CSG billing content filters (URL maps and header maps), follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg map map-name {url | header}`	Defines the CSG billing content filters (URL maps and header maps), and enters CSG map configuration mode.
Step 2	`Router(config-csg-map-header)# match protocol protocol header header-name [value pattern]`	Specifies a header match pattern for a CSG billing map.
Step 3	`Router(config-csg-map-url)# match protocol protocol [method method] url pattern`	Specifies a URL match pattern for a CSG billing map.

Note For WAP, the CSG supports URL maps, but not header maps.

Header Maps

You can specify more than one **match** command in CSG header map configuration mode to specify multiple header match expressions for a given header map:

- You can configure more than one **match header** command in a given header map, but they must reference different headers.

  For example, the following is a valid configuration, because the first **match header** command references header **Host** and the other references header **User-Agent**:

```
ip csg map HDR1
 match header Host value www.cisco.com
 match header User-Agent valuemyagent
```

  But the following is not a valid configuration, because both **match header** commands reference header **Host**:

```
ip csg map HDR1
 match header Host valuewww.cisco.com
 match header Host valuemy.cisco.com
```

- If the header matches *all* of the header match expressions, then the match is TRUE and the flows are processed by the CSG accounting services, unless another map associated with this policy matches FALSE.
- If the header *does not* match *even one* of the header match expressions, then the match is FALSE and the flows are not processed by the CSG accounting services, even if other maps for this policy match TRUE.
- The CSG treats each header match pattern as a double-wildcard match, which means that a header match pattern that includes even a single wildcard, such as **match header host* 1.2.3.4**, is treated as a triple-wildcard match. The more wildcard matches you use, the fewer header maps and header

match patterns the CSG can handle, depending on your configuration. Therefore, to optimize the performance of the CSG, minimize the number of header match patterns that are applied to a CSG content configuration, and minimize the number of wildcards used in header match patterns.

- The header match expressions are case-sensitive. For example, if you define the following header match expression:

  match header host1 value *.2.*.44

  but the actual HTTP header keyword is **HOST1**, the header *does not* match the header match expression, the match is FALSE, and the flow is not processed by the CSG accounting services.

The following example shows how to specify header match patterns for map HDR1. In this example, the header match is TRUE *only* for host **www.cisco.com** and user agent **myagent**. Any other combination of host and IP address matches FALSE:

```
ip csg map HDR1
 match header Host value www.cisco.com
 match header User-Agent value myagent
```

URL Maps

You can use more than one **match** command in CSG URL map configuration mode to specify multiple URL match expressions for a URL map:

- If the URL matches *any* of the URL match expressions, then the match is TRUE and the flows can be processed by the CSG accounting services, unless another map associated with this policy matches FALSE.
- If the URL *does not* match any of the URL match expressions, then the match is FALSE and the flows are not processed by the CSG accounting services, even if other maps for this policy match TRUE.
- The URL match expressions are case-sensitive. For example, if you define the following URL match expression:

  match protocol http url http://*url-string*

  but a subscriber enters the following URL in a web browser:

  HTTP://*url-string*

  the URL *does not* match the URL match expression, the match is FALSE, and the flow is not processed by the CSG accounting services.

  Therefore, consider uppercase and lowercase combinations carefully when you create URL match expressions.

- When you configure URL match patterns for Real Time Streaming Protocol (RTSP) streams, be sure to account for trailing stream IDs in RTSP stream names. For example, URL match pattern ***.mpeg** does not match **rtsp://1.1.1.254:554/movie.mpeg/streamid=0** because the stream name has a trailing **/streamid=0**. To match such RTSP stream names, use a URL match pattern such as ***.mpeg***.
- Depending on your configuration, the CSG can handle up to 1000 single-wildcard URL match patterns (for example, ***movies** or **movies***, but not ***movies***) or up to 11 double-wildcard URL match patterns (for example, ***movies*** or **http://test.*movies.com/*.mpeg**). Double-wildcard URL match patterns are also known as *keyword URL match patterns*. If you want to use keyword URL match patterns, observe the following guidelines to optimize the performance of the CSG:
  - Minimize the number of URL match patterns that are applied to a CSG content configuration.

- Minimize the number of keyword URL match patterns that you use. In general, it is better to use multiple single-wildcard URL match patterns instead of individual keyword URL match patterns.

– Combine multiple keyword URL match patterns into a single pattern using UNIX string-matching special characters. For example, ***.movies_comedy.com/*.mpeg**, ***.movies_action.com/*.mpeg**, and ***.movies_drama.com/*.mpeg** can be combined into the following single pattern:

***.movies_(comedy|action|drama).com/*.mpeg**

And the following patterns:

***.movies_comedy.com/*.mpeg**

***.movies_action.com/*.mpeg**

***.movies_drama.com/*.mpeg**

***.clips_comedy.com/*.mpeg**

***.clips_action.com/*.mpeg**

***.clips_drama.com/*.mpeg**

can be combined into the following single pattern:

***.(movies|clips)*?*(comedy|action|drama).com/*.mpeg**

Remember that the entire pattern, including wildcards and UNIX string-matching special characters, cannot exceed 128 characters.

- When adding or changing URL match patterns, check their effect on the CSG memory:
  1. To check the status of the configuration change, enter the **show module csg status** command in privileged EXEC mode.
  2. When the status changes from PENDING (the change has not yet downloaded) to COMPLETE, SUCCESS (the change has downloaded successfully), enter the **show module csm memory** command in privileged EXEC mode. This command displays both the total memory used and the total memory available.

The following example shows how to specify URL match patterns for map MOVIES. In this example, the URL match is TRUE for ***.movies_comedy.com/*.mpeg**, for ***.movies_action.com/*.mpeg**, and for any other URLs that match the pattern:

```
ip csg map MOVIES url
 match url *.movies_(comedy|action|drama).com/*.mpeg
```

Configuring a Symbolic Weight Name

The same weight can occur in multiple rules, specified in multiple billing services. If a weight changes, and you use numeric constants for weights, each occurrence of the weight must be updated. However, if you define symbolic weight names, you need to update only a single definition for each weight. The results are a more readable configuration and price lists that are easier to manage.

The weight name is referenced in the **content** command in CSG service configuration mode.

To define a symbolic name for a CSG billing weight, follow these steps:

Command	Purpose
`Router(config)# ip csg weight weight-name weight-value`	Defines a symbolic name for a CSG billing weight, and enters CSG weight configuration mode.

The following example shows how to define a CSG weight:

```
ip csg weight DOUBLE 2
```

Configuring Advice of Charge, Filtering, and Other Per-Event Authorizations

To configure content authorization, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg service` *service name*	Configures a content billing service, and enters CSG service configuration mode.
Step 2	`Router(config-csg-service)# authorize content`	Instructs the CSG to obtain authorization from the quota server for each subscriber request for content.

The following example shows how to configure content authorization for the CSG:

```
ip csg service service-name
 authorize content
```

To define the token used for the URL-rewriting feature of AoC, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg user-group` *group name*	Creates a group of end users for which you want to generate accounting records, and enters CSG user group configuration mode.
Step 2	`Router(config-csg-group)# aoc confirmation` *token*	Configures a token for use in Advice of Charge (AoC) URL-rewriting.

The following example shows how to specify a token for AoC URL-rewriting:

```
ip csg user-group A1
 aoc confirmation ?CSG_AOC_OK
```

Configuring Quota Server Load Sharing

The CSG allows load sharing among quota servers, similar to its BMA load balancing. Multiple quota servers can be simultaneously active, and the CSG assigns a quota server to each user.

To configure quota server load sharing, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg user group` *group name*	Creates a group of end users for which you want to generate accounting records, and enters CSG user group configuration mode.
Step 2	`Router(config-csg-group)# quota activate` *number*	Assigns a quota server to each user. All quota transactions for the user are handled by the same quota server. When a quota server fails, the transactions of all users associated with that quota server are distributed among other quota servers. Range for the *number* argument is 1 through 10.

The following example shows how to define quota server load-sharing:

```
router(config)# ip csg user u1
router(config-csg-group)# quota activate 5
```

Configuring Service-Level CDR Summarization

By default, the CSG generates billing records for each transaction. This large number of records might overwhelm the charging gateway (CG) or the collector. To prevent this situation, the CSG can summarize CDRs at the service level, instead of at the transaction level.

To configure service-level CDR summarization, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg service` *service-name*	Configures a content billing service, and enters CSG service configuration mode.
Step 2	`Router(config-csg-service)# records granularity {transaction | service {bytes` *bytes* `| time` *seconds* `| bytes` *bytes* `time` *seconds*`}}`	Specifies the granularity at which billing records (CDRs) are to be generated. For service-level CDR summarization, specify the **service** keyword.

Note To enable service-level CDR summarization in postpaid mode, you must also specify that the associated billing plan is postpaid by using the **mode postpaid** command in CSG billing configuration mode.

Service-level CDRs are generated only for subscribers with entries in the CSG User Table entry. If a subscriber does not have an entry in the User Table, the CSG generates transaction-level CDRs.

If there are no quota servers configured on the CSG, and you want to use service-level CDRs in a postpaid environment (that is, all users are postpaid), you can configure a single postpaid billing plan and assign all users to that billing plan. In the following example, all postpaid users are automatically assigned to billing plan EVERYBODY:

```
ip csg map SPORTS url
  match protocol http url http://www.nhl.com/*
!
ip csg map MOVIES url
  match protocol http url http://www.hollywood.com/*
!
ip csg policy SPORTS
  accounting type http
  url-map SPORTS
!
ip csg policy MOVIES
  accounting type http
  url-map MOVIES
!
ip csg content HTTP
  ip any tcp 80
  policy SPORTS
  policy MOVIES
  inservice
!
ip csg service SPORTS
  content HTTP policy SPORTS
  records granularity service byte 128000
!
ip csg service MOVIES
  content HTTP policy MOVIES
  records granularity service byte 128000
!
ip csg billing EVERYBODY
  mode postpaid
  service SPORTS
  service MOVIES
```

Configuring Quota Server Reauthorization

After the CSG receives a grant of zero quadrans in a Service Authorization Response, the CSG waits for an interval of time before it requests quota in a Service Reauthorization Request. To configure the initial minimum interval before the CSG sends a Service Reauthorization Request, follow these steps:

	Command	Purpose
Step 1	`Router(config)# module csg` *slot*	Enters module CSG configuration mode for a specified slot.
Step 2	`Router(config-csg-module)# variable CSG_ZERO_QUOTA_TIMEOUT_INIT` *timeout*	Sets the initial timeout for reauthorization after a quota grant of zero.

For each consecutive grant of zero quadrans in a Service Authorization Response from the quota server, the CSG doubles the retry timeout. If the quota server grants any value for quota greater than zero in a Service Authorization Response, the CSG uses the initial value for retry interval after the next zero quota grant.

Note Service authorization messages have a usage of zero for RTSP traffic.

A quota push can provide a zero grant and cause a reauthorization wait of CSG_ZERO_QUOTA_TIMEOUT_INIT.

To configure the maximum retry timeout value, follow these steps:

	Command	Purpose
Step 1	`Router(config)# module csg slot`	Enters module CSG configuration mode for a specified slot.
Step 2	`Router(config-csg-module)# variable CSG_ZERO_QUOTA_TIMEOUT_MAX timeout`	Sets the maximum timeout for reauthorization after a quota grant of zero.

Note If the INIT value is greater than the MAX value, the MAX value is used as the minimum retry interval and the INIT value is ignored.

To configure the maximum values for the threshold of available quota for sending a Service Reauthorization Request, follow these steps:

	Command	Purpose
Step 1	`Router(config)# module csg slot`	Enters module CSG configuration mode for a specified slot.
Step 2	`Router(config-csg-module)# variable CSG_BASIS_BYTE_LOW_QUOTA_MAX max_threshold`	Sets the maximum value for the available quota threshold that triggers reauthorization for duration-based billing (**basis second**).
Step 3	`Router(config-csg-module)# variable CSG_BASIS_FIXED_LOW_QUOTA_MAX max_threshold`	Sets the maximum value for the available quota threshold that triggers reauthorization for fixed-cost billing (**basis fixed**).

The CSG determines the reauthorization thresholds as follows:

- For duration-based billing (**basis second**), the threshold is the smallest of the following values:
  - CSG_BASIS_BYTE_LOW_QUOTA_MAX
  - last_quota_grant /4
  - 32 KB
- For fixed-cost billing (**basis fixed**), the threshold is the smallest of the following values:
  - CSG_BASIS_FIXED_LOW_QUOTA_MAX
  - last_quota_grant /4

Protocol-Specific Configuration Details

This section provides information about the following tasks:

- Configuring WAP and WSP Support, page 3-29
- Configuring the CSG SMTP and POP3 Billing, page 3-32

- Configuring RTSP Billing, page 3-33
- Blocking Ports, page 3-33
- Configuring Connection Duration Billing, page 3-34
- Enabling Passthrough Mode for a Service, page 3-34
- Configuring SNMP Timers, page 3-35
- Configuring the Idle Content Timer for UDP and WAP 1.x, page 3-35

Configuring WAP and WSP Support

The CSG can intercept wireless application protocol (WAP) traffic and generate reports that include contextual WAP information and counts of the bytes transferred. This feature supports both prepaid and postpaid billing. This section provides the following information:

- Incomplete WAP Transactions, page 3-29
- Multimedia Messaging Service, page 3-29
- Configuring the CSG to Monitor and Generate WAP Reports, page 3-30
- Configuring Connection-Oriented and Connectionless WAP, page 3-30
- Prepaid Support, page 3-30
- Redirect, page 3-31
- Disabling Prepaid MMS Billing, page 3-32

Incomplete WAP Transactions

When the internal session representing a WAP flow for the CSG expires (because of inactivity or receipt of a WAP DISCONNECT packet), any outstanding elements in the WAP transaction queue are reported. These outstanding elements are transactions that were not completed. Examples include a GET request for which a full REPLY was not received, and a segmented POST or PUSH that was incomplete (missing a segment). In such cases, the incomplete flag is set on the Wireless Transaction Protocol (WTP) Info Tag-Length-Value (TLV) in the WAP statistics record. The record reports the Wireless Session Protocol (WSP) PDU type, WTP transaction class, WTP transaction ID, and the number of IP bytes transferred during the attempted transaction.

Multimedia Messaging Service

The CSG differentiates Multimedia Messaging Service (MMS) traffic running over WAP from other WAP traffic by inspecting the Wireless Session Protocol (WSP) Content Type. If MMS prepaid charging is disabled, all MMS traffic flows even when non-MMS, WAP traffic is blocked because of insufficient quota. Postpaid reports for MMS are generated as for all WAP traffic.

Typically, several WAP packets are exchanged during a transaction before the WSP Content Type can be identified. When prepaid WAP with free MMS is configured, some packets still flow (even if a user has insufficient quota) in order to identify the WSP Content Type. But the transaction does not complete, and the user does not receive content if he or she has insufficient quota for a non-MMS, WAP request.

It is not always possible to determine the WSP Content Type for incomplete transactions. In these instances, no quota is deducted for prepaid users.

Configuring the CSG to Monitor and Generate WAP Reports

To enable the CSG to monitor and generate WAP traffic reports, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg policy policy-name`	Defines a policy for qualifying flows for the CSG accounting services, and enters CSG policy configuration mode.
Step 2	`Router(config-csg-policy)# accounting type wap {connection-oriented | connectionless} [customer-string string value]`	Defines the accounting type and a customer string for all flows that comply with a CSG billing policy.

The following example shows how to enable the CSG to monitor and generate WAP traffic reports:

```
ip csg policy WAP_CLT_POLICY
 accounting type wap connection-oriented customer-string to_wap_client
```

Configuring Connection-Oriented and Connectionless WAP

The **wap connection-oriented** and **wap connectionless** accounting types specify how the WAP traffic for a port is to be interpreted. To configure **wap connection-oriented** accounting or **wap connectionless** accounting, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg policy policy-name`	Defines a policy for qualifying flows for the CSG accounting services, and enters CSG policy configuration mode.
Step 2	`Router(config-csg-policy)# accounting type wap {connection-oriented | connectionless} [customer-string string]`	Defines the accounting type and a customer string for all flows that comply with a CSG billing policy.

The following example shows how to define both connection-oriented and connectionless WAP accounting types:

```
ip csg policy WSP_CON_P
 accounting type wap connection-oriented

ip csg policy WAP_NOCON_P
 accounting type wap connectionless

ip csg content WAP_CON
 ip any udp 9201
 policy WAP_CON_P

ip csg content WAP_CONLESS
 ip any udp 9200
 policy WAP_NOCON_P
```

Prepaid Support

Some upstream WAP browsing traffic occurs because the CSG must inspect the reply before determining whether the traffic is an MMS transaction. However, the downstream WAP browsing replies are discarded if quota is depleted.

Control information is charged against quota for non-MMS transactions. WSP PDU types SUSPEND and RESUME are never charged against quota.

Redirect

The CSG can redirect client flows to an alternate IP address or URL when a client's quota is exhausted. Once configured, the CSG redirects client requests to another server that informs the user that the quota has been exceeded and that describes any appropriate actions to take.

To configure the redirect option, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg user-group group-name`	Creates a group of end users for which you want to generate accounting records, and allows you to enter CSG user group configuration mode.
Step 2	`Router(config-csg-group)# redirect nat ip-address`	Redirects NAT client flows to an alternate IP address when the client's quota is exhausted.
Step 3	`Router(config-csg-group)# redirect wap url`	Redirects WAP client flows to an alternate URL when the client's quota is exhausted.
Step 4	`Router(config-csg-group)# redirect http url`	Redirects HTTP client flows to an alternate URL when the client's quota is exhausted.

WAP redirect requires configuration of a policy and service so that clients who have exhausted their quotas can access the server specified in the redirect URL.

The following example shows how to define the redirect option for WAP and how to allow redirected WAP traffic to pass without charge:

```
ip csg user-group A1
 database 10.18.12.214 3311
 radius key secret-key
 quota local-port 7788
 redirect wap http://www.topoff.com
 quota server 10.10.1.203 7777 1
ip csg map TOPOFF url
 match protocol http url http://www.topoff.com*
!
ip csg policy URL_TOPOFF
 accounting type wap connection-oriented customer-string topoff
 url-map TOPOFF
!
ip csg content WAP_WTP_CONTENT
 ip any udp 9201
 policy URL_TOPOFF
 inservice
!
ip csg weight ZERO 0
!
ip csg service FREE
 content WAP_WTP_CONTENT policy URL_TOPOFF weight ZERO
```

Disabling Prepaid MMS Billing

By default the CSG treats MMS traffic like any other WAP traffic and generates prepaid and postpaid WAP statistics reports for it. The content type distinguishes it as MMS traffic. You can disable MMS prepaid billing by performing the following task:

	Command	Purpose
Step 1	`Router(config)# ip csg service service-name`	Configures a content billing service, and enters CSG service configuration mode.
Step 2	`Router(config-csg-service)# basis byte {ip exclude mms | fixed exclude mms}`	Specifies the billing basis for a CSG content billing service. This example illustrates how to exclude prepaid billing of MMS content for volume- or fixed-basis users.
Step 3	`Router(config-csg-service)# content content-name policy policy-name`	Configures content as a member of a CSG billing service, identifies a policy to apply to this content, and optionally assigns a weight to this content.

The following example shows how to disable MMS traffic from prepaid volume billing:

```
ip csg service SERVIN_WAP
 basis byte ip exclude mms
 content WAP_CLIENT policy WAP_CLT_POLICY
 content WAP_WSP_SRV policy WAP_SRV_POLICY
 content WAP_WTP_SRV policy WAP_SRV_POLICY
```

Note You can also use the **basis fixed exclude mms** command to disable prepaid billing for fixed-basis billing.

Configuring the CSG SMTP and POP3 Billing

The CSG can report Simple Mail Transfer Protocol (SMTP) and Post Office Protocol, version 3 (POP3) data records. To configure SMTP or POP3 billing on the CSG, follow these steps:

	Command	Purpose
Step 1	`Router(config)# ip csg policy policy-name`	Defines a policy for qualifying flows for the CSG accounting services, and enters CSG policy configuration mode.
Step 2	`Router(config-csg-policy)# accounting type [smtp | pop3] [customer-string string]`	Defines the accounting type and a customer string for all flows that comply with a CSG billing policy.

The following example shows how to enable the reporting of SMTP and POP3 data records on the CSG:

```
ip csg policy SMTP
 accounting type smtp

ip csg policy POP3
 accounting type pop3

ip csg content SMTP
 ip any tcp 25
 policy SMTP
 inservice
```

```
ip csg content POP3
 ip any tcp 110
 policy POP3
 inservice
```

Configuring RTSP Billing

RTSP billing correlates all the streams that are associated with an RTSP session, and reports application-level information (for example, filename) to the billing system.

To configure RTSP billing on the CSG, enter the following command in CSG policy configuration mode:

Command	Purpose
`Router(config-csg-policy)# accounting type rtsp [customer-string string]`	Defines the accounting type as RTSP, and optionally a customer string for all flows that comply with a CSG billing policy. Prepaid service matches are based on the IP address and port number of the control connection to the RTSP server IP.

The following example shows how to configure RTSP billing:

```
ip csg policy RTSP
 accounting type rtsp

ip csg content RTSP
 ip any tcp 554
 policy RTSP
 inservice
```

When you configure RTSP billing, keep the following considerations in mind:

- The CSG supports only port 554 for RTSP billing.
- RealPlayer clients ignore the explicit definition of port 554 in the URL, and attempt to connect to ports 554, 7070, 80, and 8080. Many other streaming media servers also listen on ports 7070, 80, and 8080. For HTTP transport, if the media streams from any port other than port 554 (such as port 7070, 80, or 8080), the CSG does not bill the stream as RTSP. Therefore, for RTSP billing, you must block TCP and HTTP connections to the server network on ports 7070, 80, and 8080. For more information about blocking ports, see the "Blocking Ports" section on page 3-33.
- When using HTTP as the transport for RTSP, the control connection might time out, causing the stream to become unresponsive. This occurs because the client opens two TCP connections, one for the main content and one for control. The client uses the control connection sparingly, which can cause the the connection to time out. To prevent this problem, ensure that the idle content timer has a duration of at least 60 seconds (the default setting is 3600 seconds). The description of the **idle** command provides more information on setting the idle content timer.

  This is not an issue when using UDP or TCP as the transport.

Blocking Ports

To block a port, you must configure a content that matches the connection to the server network and configure a policy that sends transactions to a false next-hop IP address, as shown in the following example:

```
ip csg policy RTSP
 accounting type rtsp
!
ip csg policy RTSP-BLOCK
 next-hop 10.10.10.1
!
ip csg content BLOCK7070
 ip 1.1.1.0 255.255.255.0 tcp 7070
 policy RTSP-BLOCK
 inservice
!
ip csg content BLOCK80
 ip 1.1.1.0 255.255.255.0 tcp 80
 policy RTSP-BLOCK
 inservice
!
ip csg content BLOCK8080
 ip 1.1.1.0 255.255.255.0 tcp 8080
 policy RTSP-BLOCK
 inservice
!
ip csg content RTSPCONTSERVER
 ip 1.1.1.0 255.255.255.0 tcp 554
 idle 50
 replicate
 policy RTSP
 inservice
```

Configuring Connection Duration Billing

Connection Duration Billing enables the CSG to deduct quota based on the time that a user is logged on to the IP network.

To configure the Connection Duration Billing feature on the CSG, specify the following commands in CSG service configuration mode:

	Command	Purpose
Step 1	`Router(config-csg-service)# basis second connect [exclude mms]`	Specifies Connection Duration Billing for a CSG content billing service. **Note** When you change the basis for a service, you must take the content out of service.
Step 2	`Router(config-csg-service)# activation [automatic | user-profile]`	Specifies the activation mode for a Connection Duration service.

Enabling Passthrough Mode for a Service

To enable passthrough mode for a service, enter the following command in CSG service configuration mode:

Command	Purpose
`Router(config-csg-service)# passthrough` *quota-grant*	Enables passthrough mode for a service.

The following example specifies that the CSG grants 65,535 quadrans of quota to the service NAME each time the service runs low on quota:

```
ip csg service NAME
 passthrough 65535
```

Configuring SNMP Timers

The CSG enables you to configure SNMP timers for lost CSG records.

To configure an SNMP timer and to enter CSG SNMP timer configuration mode, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# ip csg snmp timer {agent | quota-server} [interval]`	Defines SNMP timers for lost CSG records, and enters CSG SNMP timer configuration mode.

The following example defines a 300-second CSG SNMP agent timer:

```
ip csg snmp timer agent 300
```

Configuring the Idle Content Timer for UDP and WAP 1.x

To configure an idle content timer, enter the following command in CSG content configuration mode:

Command	Purpose
`Router(config-csg-content)# idle duration`	Specifies the minimum amount of time that the CSG maintains an idle content connection.

The following example shows how to configure a 120-second idle timer for the CSG content MOVIES_COMEDY:

```
ip csg content MOVIES_COMEDY
 idle 120
```

The CSG tracks usage on a per-session basis. UDP protocols do not have an end-of-session indicator and the sessions simply idle out. For UDP and WAP 1.x, setting the content idle timer to a low value (for example, 30 seconds) allows the CSG to quickly recognize that a session has ended and to generate billing records accordingly. Other service-level features of the CSG that count sessions (such as passthrough mode and service-level CDRs) are similarly affected by the content idle timer setting.

Other Configuration Tasks

The following sections provide additional information on configuring the CSG:

- Configuring the CSG and PSD, page 3-36
- Configuring VLANs, page 3-36
- Configuring Layer 2–Adjacent Devices, page 3-39

Configuring the CSG and PSD

The configuration tasks for establishing communication between the CSG and the Cisco Persistent Storage Device (PSD) involve steps that are beyond the scope of this chapter. For specific information on configuring the CSG and the PSD, see Appendix A, "Configuring the Cisco Persistent Storage Device for the CSG."

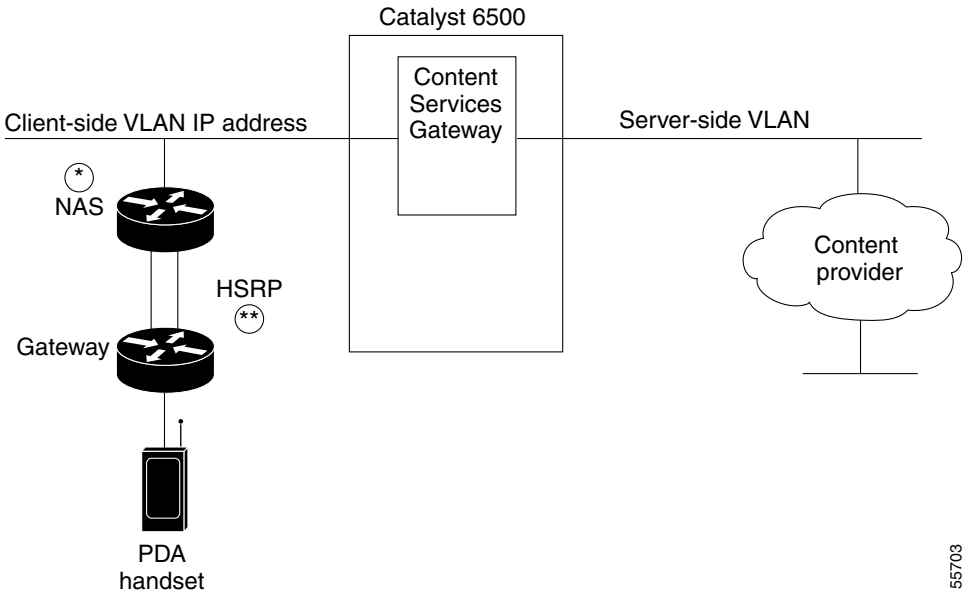
Configuring VLANs

Clients and servers communicate through the CSG using Layer 2 and Layer 3 technology in a specific VLAN configuration. Clients connect to the client-side VLAN, and servers connect to the server-side VLAN. Servers and clients exist on different subnets. Servers can also be located one or more Layer 3 hops away, and they can connect to the server-side VLAN through routers. This section describes how to configure VLANs for the CSG.

A client sends a request to one of the module's server addresses. The CSG extracts the URL—if applicable—and records the statistics. When properly configured, the CSG records statistics for flows in both directions. When a connection ends, the CSG builds an accounting record and sends it to the BMA.

When you install the CSG in a Catalyst 6500 series switch, you must configure both a client-side VLAN and a server-side VLAN. (See Figure 3-1.)

Figure 3-1 Configuring VLANs



*Any router that is configured as a client-side gateway, or as a next-hop router for servers more than one hop away, must have Internet Control Message Protocol (ICMP) redirects disabled. The CSG does not perform a Layer 3 lookup to forward traffic; the CSG cannot act upon ICMP redirects.

** You can configure up to 7 gateways per VLAN for up to 256 VLANs, and up to 224 gateways for the entire system. If a Hot Standby Router Protocol (HSRP) gateway is configured, the CSG uses 3 of the 224 gateway entries because traffic can come from both the virtual and physical MAC addresses of the HSRP group. (See the "HSRP Configuration Overview" section on page 4-8.)

Note You must configure VLANs on the Catalyst 6000 series switch or Cisco 7600 series router *before* you configure VLANs for the CSG. VLAN IDs must be the same for both the switch and the module.

You must create both a client-side VLAN and a server-side VLAN:

- Configuring a Client-Side VLAN, page 3-37
- Configuring a Server-Side VLAN, page 3-37
- Associating a Table Name with a VLAN, page 3-39

Configuring a Client-Side VLAN

To configure a client-side VLAN, follow these steps:

	Command	Purpose
Step 1	`Router(config-csg-module)# vlan vlan-id client [vlan-name]`	Configures a client-side VLAN and enters the client VLAN mode. **Note** Do not use VLAN 1 as a client-side VLAN for the CSG.
Step 2	`Router(config-csg-vlan-client)# ip address ip-address netmask`	Configures an IP address to the CSG used by probes and Address Resolution Protocol (ARP) requests on this particular VLAN.
Step 3	`Router(config-csg-vlan-client)# gateway ip-address`	Configures the gateway IP address.

The following example shows how to configure a client-side VLAN:

```
Router(config-module-csg)# vlan 130 client
Router(config-csg-vlan-client)# ip address 123.44.50.6 255.255.255.0
Router(config-csg-vlan-client)# gateway 123.44.50.1
Router(config-csg-vlan-client)# exit
```

Configuring a Server-Side VLAN

To configure a server-side VLAN, follow these steps:

	Command	Purpose
Step 1	`Router(config-csg-module)# vlan vlan-id server [vlan-name]`	Configures a server-side VLAN and enters the server VLAN mode. **Note** Do not use VLAN 1 as a server-side VLAN for the CSG.
Step 2	`Router(config-csg-vlan-server)# ip address ip-address netmask`	Configures an IP address for the server VLAN.

	Command	Purpose
Step 3	`Router(config-csg-vlan-server)# alias ip-address netmask`	(Optional) Configures multiple IP addresses (aliases) to the CSG as alternate gateways for the real server. An alias is required in the redundant configuration.
Step 4	`Router(config-csg-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 CSG. **Note** If you are adding a new route to an existing gateway, the new route might not take effect until you remove the gateway and reconfigure it to clear the gateway cached entries.

The following example shows how to configure a server-side VLAN:

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

Associating a Table Name with a VLAN

Interface awareness enables the CSG to distinguish between users and sessions that share the same IP address on different VLANs (that is, users and sessions with overlapping IP addresses). Interface awareness requires that each VLAN be associated with a table name.

To associate a table name with a VLAN, enter the following command in module CSG VLAN configuration mode:

Command	Purpose
`Router(config-csg-vlan-client)# table` *table-name* or `Router(config-csg-vlan-server)# table` *table-name*	Associates a table name with a VLAN.

Configuring Layer 2–Adjacent Devices

If a CSG receives a packet with a Layer 2 address that it does not recognize, from a device that has a Layer 3 address that is not on the same IP subnet as the CSG, the CSG drops the packet. This dropping of packets can be a problem if Layer 2-adjacent devices are performing redundancy (for example, HSRP firewalls).

To avoid this dropping of packets, configure static routes on the CSG that point to the IP addresses on the interfaces of the adjacent devices or firewalls. For example, if the CSG is Layer 2–adjacent to two firewalls, and the IP addresses on those firewalls are 1.1.1.5 and 1.1.1.6, configure the following on the CSG:

route *IP address not-in-use on the network* 255.255.255.255 gateway 1.1.1.5

route *IP address not-in-use on the network* 255.255.255.255 gateway 1.1.1.6

This configuration causes the CSG to spawn an ARP for 1.1.1.5 and 1.1.1.6 so that it has an ARP entry in its ARP cache for both firewalls. In the event of a failover, the packets received from the now-active firewall have a source MAC that is in the ARP cache of the CSG.

Configuration Examples

This section includes the following examples:

- Sample CSG Billing Rules, page 3-40
- Simple Postpaid Billing Configuration Example, page 3-43
- Basic WAP Configuration Example, page 3-43
- Redirect to Top-Off Server Configuration Example, page 3-44
- Free MMS Transactions Configuration Example, page 3-45

- Differentiating MMS Over WAP 2.0 Example, page 3-47
- Pricing by Quota Server Configuration Example, page 3-48
- Differentiating Prices Configuration Example, page 3-49
- Reducing the Number of Services Configuration Example, page 3-50
- Interface Awareness Example, page 3-51

Sample CSG Billing Rules

Table 3-1 lists sample CSG billing rules.

Table 3-1 Sample CSG Billing Rules

Content Configuration	Service and Billing Basis	Quadrans per Unit
IP/Netmask = 1.2.3.4/24 Protocol/Port Number = TCP/80 HostName = *.books-co-inc.com URL = *.jpg	Service = BillByVolume Basis = TCP Volume	1
IP/Netmask = 1.2.3.4/24 Protocol/Port Number = TCP/80 HostName = *.books-co-inc.com URL = *freecontent*	Service = BillPerClick Basis = Constant	0
IP/Netmask = 1.2.3.4/24 Protocol/Port Number = TCP/80 HostName = *.advt-co.com URL = *	Service = Advertisements Basis = Constant	-1
IP/Netmask = 198.133.219.0/24 Protocol/Port Number = TCP/80 HostName = *bigcorp* URL = *	Service = Corporate Basis = Constant	0
IP/Netmask = 0.0.0.0/0 Protocol/Port Number = TCP/80 HostName = * URL = *	Service = Internet Basis = IP Volume	1

The following example shows how to configure these CSG billing rules:

```
ip csg user-group U1
 entries max 10000
 radius key cisco
 radius acct-port 23385
 radius userid User-Name
 quota local-port 4095
 quota server 20.20.50.13 3386 5
```

```
 quota server 20.20.50.130 3386 6
 quota server 20.20.52.13 3386 7
!
ip csg accounting CSGBILL
 user-group U1
 records max 2000
 agent activate 2 sticky 30
 records intermediate bytes 50000
 agent 9.15.72.5 3386 2
 agent 10.76.86.2 3386 5
!
 agent 20.20.50.131 3386 8
 inservice
!
ip csg map ADVERTISEMENTS header
 match header Host header-value *.advt-co.com
!
ip csg map ALLHOSTS header
 match header Host header-value *
!
ip csg map BOOKS header
 match header Host header-value *.books-co-inc.com
!
ip csg map CORPORATE header
 match header Host header-value *bigcorp*
!
ip csg map ALLURLS url
 match url *
!
ip csg map BOOKFREE url
 match url *freecontent*
!
ip csg map JPGS url
 match url *.jpg
!
ip csg map GIF url
 match url *.gif
!
ip csg policy ADVERTISEMENTS
 accounting type http
 url-map ALLURLS
 header-map ADVERTISEMENTS
!
ip csg policy BOOKFREE
 accounting type http
 url-map BOOKFREE
 header-map BOOKS
!
ip csg policy BOOKSALES
 accounting type http
 url-map JPGS
 header-map BOOKS
!
ip csg policy CORPORATE
 accounting type http
 url-map ALLURLS
 header-map CORPORATE
!
ip csg policy INTERNET
 accounting type http
 url-map ALLURLS
 header-map ALLHOSTS
!
ip csg content ADVERTISEMENTS
```

```
 ip 1.2.5.0 255.255.255.0 tcp 80
 policy ADVERTISEMENTS
 inservice
!
ip csg content BOOKS
 ip 1.2.3.0 255.255.255.0 tcp 80
 policy BOOKSALES
 policy BOOKFREE
 inservice
!

ip csg content CORPORATE
 ip 198.133.219.0 255.255.255.0 tcp 80
 policy CORPORATE
 inservice
!
ip csg content INTERNET
 ip any tcp 80
 policy INTERNET
 inservice
!
ip csg ruleset R1
 content ADVERTISEMENTS
 content BOOKS
 content CORPORATE
 content INTERNET
!
ip csg weight FREE 0
ip csg weight PAYBACK -1
!
ip csg service BILLPERCLICK
 basis fixed
 content ADVERTISEMENTS policy ADVERTISEMENTS weight PAYBACK
 content BOOKS policy BOOKSALES
 content BOOKS policy BOOKFREE weight FREE
 content CORPORATE policy CORPORATE weight FREE
!
ip csg service BILLBYVOLUME
 basis byte tcp
 content BILLBYVOLUME policy BILLBYVOLUME
!
ip csg service BILLBYIPVOLUME
 basis byte
 content INTERNET policy INTERNET
!
ip csg billing PLAN1
 service BILLPERCLICK
 service BILLBYVOLUME
 service BILLBYIPVOLUME
!

module ContentServicesGateway 5
 vlan 30 client AUCTION_HOUSE
  ip address 123.44.50.6 255.255.255.0
  gateway 123.44.50.1
!
 vlan 40 server
  ip address 123.46.50.6 255.255.255.0
!
 ruleset R1
 accounting CSGBILL
```

Simple Postpaid Billing Configuration Example

The following example shows a simple postpaid billing CSG configuration:

```
ip csg policy POLICY1
 accounting type http
!
ip csg content MOVIES_COMEDY
 ip 172.18.45.0/24 tcp 8080
 policy POLICY1
 inservice
!
ip csg content AUCTION_HOUSE
 ip 216.32.120.0/24 tcp 8080
 policy POLICY1
 vlan AUCTION_HOUSE
 inservice
!
ip csg content WAKETECH
 ip 48.33.0.0/16 tcp 80
 policy POLICY1
 inservice
!
ip csg ruleset R1
 content MOVIES_COMEDY
 content AUCTION_HOUSE
 content WAKETECH
!
ip csg user-group G1
 entries max 100000
 database 10.1.2.3 11111
 radius key secretpassword
!
ip csg accounting A1
 user-group G1
 agent localport 3775
 agent 10.1.2.4 11112 1
 agent 10.1.2.5 11113 2
 agent activate 2
 records max 250
 inservice
!
mod csg 4
 vlan 30 client AUCTION_HOUSE
  ip address 123.44.50.6 255.255.255.0
  gateway 123.44.50.1
 vlan 40 server
  ip address 123.46.50.6 255.255.255.0
  alias 123.60.7.6 255.255.255.0
  route 123.50.0.0 255.255.0.0 gateway 123.44.50.1
 ruleset R1
 accounting A1
```

Basic WAP Configuration Example

The following example provides a basic CSG WAP configuration with the following functions:

- Charges a fixed rate for all WAP and MMS transactions for which a URL is used
- Allows requests that are not content-based (control flows) to go through for free
- Uses a single service for all traffic

```
ip csg map DEFAULT_URL url
 match protocol http url http://*
!
ip csg policy WAP_URL
 accounting type wap connection-oriented
 url-map DEFAULT_URL
!
ip csg policy WAP_CONTROL
 accounting type wap connection-oriented customer-string control_flow
!
ip csg content WAP_WTP_CONTENT
 ip any udp 9201
 idle 30
 policy WAP_URL
 policy WAP_CONTROL
 inservice
!
ip csg weight ZERO 0
!
ip csg service WAP
 basis fixed
 content WAP_WTP_CONTENT policy WAP_URL
 content WAP_WTP_CONTENT policy WAP_CONTROL weight ZERO
```

Redirect to Top-Off Server Configuration Example

The following example illustrates a WAP configuration with additions to support redirect to a top-off server. This configuration provides the following functions:

- Allows redirect requests to the top-off server to go through for free
- Defines a second service to be used only for free transactions

Note This configuration is required to allow redirect to work properly.

Users must also be authorized to use this service by the quota server.

No quota needs to be given out for this service, but a cause code of 0x04 (user authorized) must be returned for the transaction to be allowed through.

```
ip csg map TOPOFF url
 match protocol http url http://www.topoff.com*
!
ip csg map DEFAULT_URL url
 match protocol http url http://*
!
ip csg policy URL_TOPOFF
 accounting type wap connection-oriented customer-string topoff
 url-map TOPOFF
!
ip csg policy WAP_URL
 accounting type wap connection-oriented customer-string
 url-map DEFAULT_URL
!
ip csg policy WAP_CONTROL
 accounting type wap connection-oriented customer-string control_flow
!
ip csg content WAP_WTP_CONTENT
 ip any udp 9201
 idle 30
```

```
 policy URL_TOPOFF
 policy WAP_URL
 policy WAP_CONTROL
 inservice
!
ip csg weight ZERO 0
!
ip csg service WAP
 basis fixed
 content WAP_WTP_CONTENT policy WAP_URL
!
ip csg service FREE
 content WAP_WTP_CONTENT policy URL_TOPOFF weight ZERO
 content WAP_WTP_CONTENT policy WAP_CONTROL weight ZERO
```

Free MMS Transactions Configuration Example

This sections provides the following examples:

- Specific MMS Transactions, page 3-45—in which some MMS transactions are free
- All MMS Transactions, page 3-46—in which all MMS transactions are free

Specific MMS Transactions

The following example shows a WAP 1 or MMS/WAP 1.x configuration in which MMS transactions to servers mms1 and mms2 are free, while third-party MMS transactions are charged.

```
ip csg map TOPOFF url
 match protocol http url http://www.topoff.com*
!
ip csg map OUR_MMS url
 match protocol http url http://www.mms1*
 match protocol http url http://www.mms2*
!
ip csg map DEFAULT_URL url
 match protocol http url http://*
!
ip csg policy URL_TOPOFF
 accounting type wap connection-oriented customer-string topoff
 url-map TOPOFF
!
ip csg policy FREE_MMS
 accounting type wap connection-oriented customer-string free_mms
 url-map OUR_MMS
!
ip csg policy WAP_URL
 accounting type wap connection-oriented customer-string
 url-map DEFAULT_URL
!
ip csg policy WAP_CONTROL
 accounting type wap connection-oriented customer-string control_flow
!
ip csg content WAP_WTP_CONTENT
 ip any udp 9201
 idle 30
 policy URL_TOPOFF
 policy FREE_MMS
 policy WAP_URL
 policy WAP_CONTROL
 inservice
```

```
!
ip csg weight ZERO 0
!
ip csg service WAP
 basis fixed
 content WAP_WTP_CONTENT policy WAP_URL
!
ip csg service FREE
 content WAP_WTP_CONTENT policy URL_TOPOFF weight ZERO
 content WAP_WTP_CONTENT policy FREE_MMS weight ZERO
 content WAP_WTP_CONTENT policy WAP_CONTROL weight ZERO
```

All MMS Transactions

The following example shows a WAP 1 or MMS/WAP 1.x configuration in which all MMS transactions are free. In this example, MMS content is free for service WAP (the user must be authorized for this service).

```
ip csg map TOPOFF url
 match protocol http url http://www.topoff.com*
!
ip csg map DEFAULT_URL url
 match protocol http url http://*
!
ip csg policy URL_TOPOFF
 accounting type wap connection-oriented customer-string topoff
 url-map TOPOFF
!
ip csg policy WAP_URL
 accounting type wap connection-oriented customer-string
 url-map DEFAULT_URL
!
ip csg policy WAP_CONTROL
 accounting type wap connection-oriented customer-string control_flow
!
ip csg content WAP_WTP_CONTENT
 ip any udp 9201
 idle 30
 policy URL_TOPOFF
 policy WAP_URL
 policy WAP_CONTROL
 inservice
!
ip csg weight ZERO 0
!
ip csg service WAP
  basis fixed exclude mms
  content WAP_WTP_CONTENT policy WAP_URL
!
ip csg service FREE
  content WAP_WTP_CONTENT policy URL_TOPOFF weight ZERO
  content WAP_WTP_CONTENT policy WAP_CONTROL weight ZERO
```

Differentiating MMS Over WAP 2.0 Example

The following example assumes that the quota server and the accounting agent are already configured for the system. It also assumes that the WAP proxy can be found on port 9401 on a host that can be addressed by using a server VLAN that is configured to access the subnet 10.10.2.0/24. This example shows a working configuration for differentiating billing WAP 2.0/HTTP and MMS/WAP 2.0/HTTP.

```
ip csg map WAP2MMS_GET_MAP url
 match protocol http method GET url /wap/mms*

ip csg map WAP2MMS_POSTMAP url
 match protocol http method POST url /wap/mms*

ip csg map WAP2MMSPOSTMAPH header
 match protocol http header Content-Type header-value application/vnd.wap.mms-message

ip csg policy WAP2_MMS_GET
! match all wap2/http gets of mms
  accounting type http customer-string wap2mms-get
  url-map WAP2MMS_GET_MAP

ip csg policy WAP2_MMS_POST
! match all wap2/http posts that are mms related
! This catches handset-initiated MMS sends and acknowledgements of
! network-initiated MMS pushes.
  accounting type http customer-string wap2mms-post
  header-map WAP2MMSPOSTMAPH  ! recommended
! or
! url-map WAP2MMS_POSTMAP  ! optional
! The header-map catches MMS even when it goes to an unknown URL,
! so it is recommended over the url-map.

ip csg policy WAP2
! You might choose to differentiate non-MMS wap2 get/posts and URLs/headers
! here, if relevant. In this case, we just label all remaining traffic as
! wap2.
  accounting type http customer-string wap2

ip csg content WAP2
! 10.10.2.0 255.255.255.0 represents the network where WAP 2.0 Proxies are
! located. Port 9401 is the port the WAP 2.0 Proxies are configured to use.
 ip 10.10.2.0 255.255.255.0 tcp 9401
 policy WAP2_MMS_GET
 policy WAP2_MMS_POST
 policy WAP2
 inservice

! Adjust these to change the pre-paid weight associated with each flow:
ip csg weight WEIGHT_WAP2 3
ip csg weight WEIGHT_WAP2GET 1
ip csg weight WEIGHT_WAP2POST 2

ip csg service WAP2MMSGET
 basis fixed
 idle 10000
 content WAP2 policy WAP2_MMS_GET weight WEIGHT_WAP2GET

ip csg service WAP2MMSPOST
 basis fixed
 idle 10000
 content WAP2 policy WAP2_MMS_POST weight WEIGHT_WAP2POST
```

```
ip csg service WAP2
 basis fixed
 idle 10000
 content WAP2 policy WAP2 weight WEIGHT_WAP2

ip csg ruleset R
! other contents
 content WAP2

ip csg billing BILL1
 service WAP2MMSGET
 service WAP2MMSPOST
 service WAP2
```

Pricing by Quota Server Configuration Example

The following example shows a CSG configuration in which the quota server performs all pricing. In this example:

- Assume that User X has $10.00 in his account.
- There are two types of content:
  - C1—This content is billed per object (for example, URL GET); each object costs $0.01.
  - C2—This content is billed per byte; each kilobyte costs $0.01.
- The quota server controls each object transaction for content C1.
- The quota server controls all the pricing.

```
ip csg content C1
 policy P1
 inservice
!
ip csg content C2
 policy P2
 inservice
!
ip csg service PERCLICK
 basis fixed
 content C1 policy P1
!
ip csg service PERBYTE
 basis byte ip exclude mms
 content C2 policy P2
!
ip csg billing REGULAR
 service PERCLICK
 service PERBYTE
```

When User X, with a subscription to billing plan REGULAR, tries to access content that matches C1, the CSG tries to download quota for User X for service PERCLICK.

The quota server borrows money from User X's $10.00, and returns some quadrans to the CSG. Each quadran is good for one object download, or one click. If the quota server is configured for the CSG to query for each click, it can choose to send just one quadran at a time, so that the CSG queries the quota server each time. However, if the quota server is configured to grant $2.00 worth of quadrans to the CSG in one shot, it can send 200 quadrans to the CSG, which the CSG keeps using for User X's access to C1.

When User X tries to access content that matches C2, the CSG makes another request to the quota server to get User X's quota for C2. C2 is billed per IP byte. The quota server borrows another $5.00 from User X's account, and sends 500000 quadrans to the CSG. As User X continues to access C2, his traffic is metered for volume. For each byte, the CSG deducts one quadran.

Differentiating Prices Configuration Example

The following example extends the previous example by adding a content type that is priced differently. In this example:

- Assume that User X has $10.00 in his account.
- There are three types of content:
  - C1—This content is billed per *.jpg file, where each JPG file costs $0.01.
  - C2—This content is billed per byte, where each kilobyte costs $0.01.
  - C3—This content is billed per *.mp3 file, where each MP3 file costs $0.05.
- The quota server controls each object transaction for content C1.
- The quota server controls all the pricing.

This configuration requires an additional service type, MP3, which allows the quota server to price object downloads (clicks) differently for MP3 files.

```
ip csg content C1
 policy P1
 inservice
!
ip csg content C2
 policy P2
 inservice
!
ip csg content MP3
 policy P1
 inservice
!
ip csg service PERCLICK
 basis fixed
 content C1 policy P1
!
ip csg service PERBYTE
 basis byte ip
 content C2 policy P2
!
ip csg service MP3
 basis fixed
 content C1 policy P1
!
ip csg billing REGULAR
 service PERCLICK
 service PERBYTE
 service MP3
```

When User X tries to download an MP3 file (that is, a file that matches content type MP3), the CSG requests the MP3 quota for User X. Each download of an MP3 file costs $0.05, so the quota server borrows $1.00 from User X's account, and returns 20 quadrans to the CSG for service MP3. The CSG can use the quadrans for 20 downloads of MP3 files.

Alternatively, the quota server could send just one quadran, which is enough for only one transaction. This would force the CSG to ask for quota before each download of an MP3 file.

Reducing the Number of Services Configuration Example

The "Differentiating Prices Configuration Example" section on page 3-49 shows that you can create a new service for one type of content and differentiate its billing from other types of content.

However, with each new service, the user's quota fragments further, and traffic between the CSG and the quota server increases.

You can reduce traffic by specifying a symbolic weight on the CSG. In the following example, each MP3 download ($0.05) costs five times as much as each JPG download ($0.01). By assigning a weight of 5 to MP3 downloads, you can keep both content C1 and content MP3 under service PERCLICK, thereby reducing the overall number of services and reducing the traffic between the CSG and the quota server.

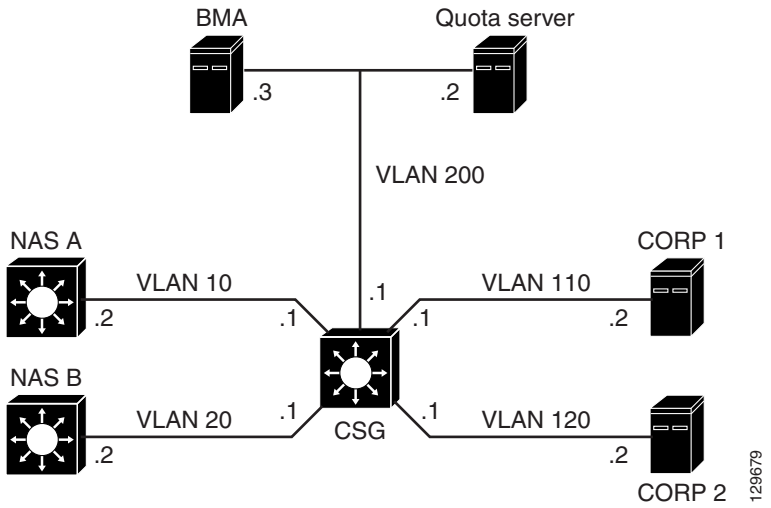
```
ip csg content C1
 policy P1
 inservice
!
ip csg content C2
 policy P2
 inservice
!
ip csg content MP3
 policy P1
 inservice
!
ip csg weight MP3 5
!
ip csg service PERBYTE
 basis byte ip
 content C2 policy P2
!
ip csg service PERCLICK
 basis fixed
 content C1 policy P1
 content MP3 policy P1 weight MP3
!
ip csg billing REGULAR
 service PERCLICK
 service PERBYTE
```

When the quota server borrows $1.00 from User X's account and sends 100 quadrans for service PERCLICK, the CSG can use the quadrans for 100 JPG files, or for 20 MP3 files, or for a mix of the two content types.

Interface Awareness Example

The following example provides a sample configuration for interface awareness.

Figure 3-2 Interface Awareness




```
ip csg user-group GROUP1
 radius userid Calling-Station-Id
 user-profile server radius pass
 quota server 10.10.200.2 3386 1
!
ip csg accounting USER-BMA1
 user-group GROUP1
 agent 10.10.200.3 3386 1
 inservice
!
ip csg policy CORP1-POLICY
 accounting type other customer-string CORP1
 next-hop 10.10.110.2
!
ip csg policy CORP2-POLICY
 accounting type other customer-string CORP2
 next-hop 10.10.120.2
!
ip csg content CORP1-CONTENT
 ip any
 vlan CORP1-CLIENT
 policy CORP1-POLICY
 inservice
!
ip csg content CORP2-CONTENT
 ip any
 vlan CORP2-CLIENT
 policy CORP2-POLICY
 inservice
!
ip csg ruleset R1
 content CORP1-CONTENT
 content CORP2-CONTENT
!
ip csg service CORP1
```

```
 content CORP1-CONTENT policy CORP1-POLICY
!
ip csg service CORP2
 content CORP2-CONTENT policy CORP2-POLICY
!
ip csg billing CORP1
 service CORP1
!
ip csg billing CORP2
 service CORP2
!
module ContentServicesGateway 9
 vlan 10 server
  name CORP1-CLIENT
  table C1
  ip address 10.10.10.1 255.255.255.0
!
 vlan 20 server
  name CORP2-CLIENT
  table C2
  ip address 10.10.20.1 255.255.255.0
!
 vlan 110 server
  name CORP1-SERVER
  table C1
  ip address 10.10.110.1 255.255.255.0
!
 vlan 120 server
  name CORP2-SERVER
  table C2
  ip address 10.10.120.1 255.255.255.0
!
 vlan 200 server
  name CSG-TO-BMA-QS
  ip address 10.10.200.1 255.255.255.0
!
 ruleset R1
 accounting USER-BMA1
 radius proxy 10.10.10.3 10.10.110.3 key cisco table C1
 radius proxy 10.10.20.3 10.10.120.3 key cisco table C2
```