



Configuring Simple Network Management Protocol (SNMP)

This chapter provides information on configuring Simple Network Management Protocol (SNMP) features of your CSS. It also provides a brief overview of SNMP, an Application Layer protocol used extensively in the communications industry. Information in this chapter applies to all CSS models except where noted.

This chapter includes the following major sections:

- [SNMP Overview](#)
- [Management Information Base \(MIB\) Overview](#)
- [Preparing to Configure SNMP on the CSS](#)
- [Defining the CSS as an SNMP Agent](#)
- [Configuring Denial of Service \(DoS\)](#)
- [Displaying the SNMP Configuration](#)
- [Managing SNMP on the CSS](#)
- [CSS SNMP Traps](#)
- [CSS MIBs](#)

SNMP Overview

SNMP is a set of network management standards for IP-based internetworks. SNMP includes a protocol, a database-structure specification, and a set of management data objects. SNMP implementations typically consist of a management application running on one or more network management systems (NMSs), and agent applications, usually executing in firmware on various network devices.

SNMP has two major standard revisions, SNMPv1 and SNMPv2. The CSS supports both SNMPv1 and SNMPv2C (SNMP version 2C), a standard Management Information Base (MIB-II) object, along with an extensive set of enterprise MIB objects. MIBs are discussed in the [“Management Information Base \(MIB\) Overview”](#) section.

This section contains the following topics:

- [Managers and Agents](#)
- [SNMP Manager and Agent Communication](#)



Note

By default, SNMP access to the CSS is enabled through the **no restrict snmp** command. For details, see the [“Preparing to Configure SNMP on the CSS”](#) section.

Managers and Agents

SNMP uses software entities called *managers* and *agents* to manage network devices:

- The *manager* monitors and controls all other SNMP-managed devices (network nodes) in the network. There must be at least one SNMP manager in a managed network. The manager is installed on a workstation somewhere in the network.
- An *agent* resides in a managed device (a network node). The agent receives instructions from the SNMP manager, and also sends management information back to the SNMP manager as events occur. The agent can reside on routers, bridges, hubs, workstations, or printers, to name just a few network devices.

There are many different SNMP management applications, but they all perform the same basic task: They allow SNMP managers to communicate with agents to monitor, configure, and receive alerts from the network devices. You can use any SNMP-compatible NMS to monitor and control a CSS.

SNMP Manager and Agent Communication

There are several ways that the SNMP manager and the agent communicate.

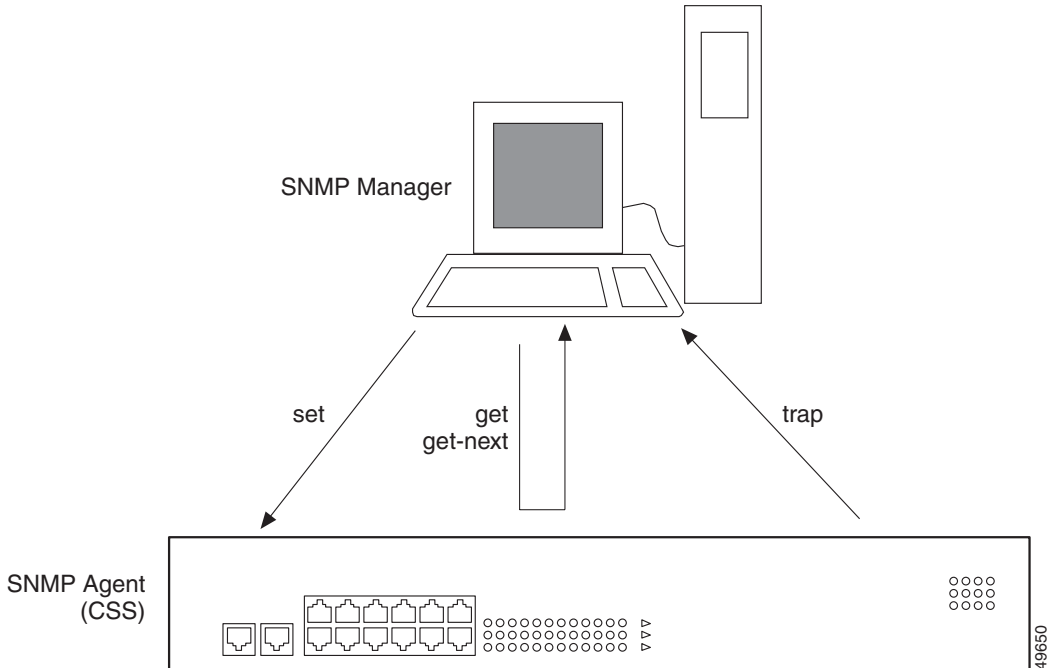
- The manager can:
 - Retrieve a value (a GET action).

The SNMP manager requests information from the agent, such as the number of users logged on to the agent device, or the status of a critical process on that device. The agent gets the value of the requested MIB object and sends the value back to the manager.
 - Retrieve the value immediately after the variable you name (a GET-NEXT action).

The SNMP manager retrieves values from within a MIB. Using the get-next function, you do not need to know the exact MIB object instance you are looking for; the SNMP manager takes the variable you name and then uses a sequential search to find the desired variables.

- Retrieve a number of values (a GET-BULK action).
The SNMP manager performs a number of get-next actions that you specify.
- Change a setting on the agent (a SET action).
The SNMP manager requests the agent to change the value of the MIB object. For example, you could run a script or an application on a remote device with a set action.
- An agent can send an unsolicited message to the manager at any time if a significant, predetermined event takes place on the agent. This message is called a *trap*. For details on SNMP traps (and associated MIB objects) supported by the CSS software, see the “[CSS SNMP Traps](#)” section.
When a trap condition occurs, the SNMP agent sends an SNMP trap message to the device specified as the *trap receiver* or *trap host*. The SNMP Administrator configures the trap host (usually the SNMP management station) to perform the action needed when a trap is detected. [Figure 12-1](#) illustrates SNMP manager and agent communication.

Figure 12-1 SNMP Manager and Agent Interaction



Management Information Base (MIB) Overview

SNMP obtains information from the network through a Management Information Base (MIB). The MIB is a database of code blocks called *MIB objects*. Each MIB object controls one specific function, such as counting how many bytes are transmitted through an agent's port. The MIB object comprises *MIB variables*, which define the MIB object name, description, default value, and so forth.

The collection of MIB objects is structured hierarchically. The MIB hierarchy is referred to as the *MIB tree*. The MIB tree is defined by the International Standards Organization (ISO). The MIB is installed on the SNMP manager and is present within each agent in the SNMP network.

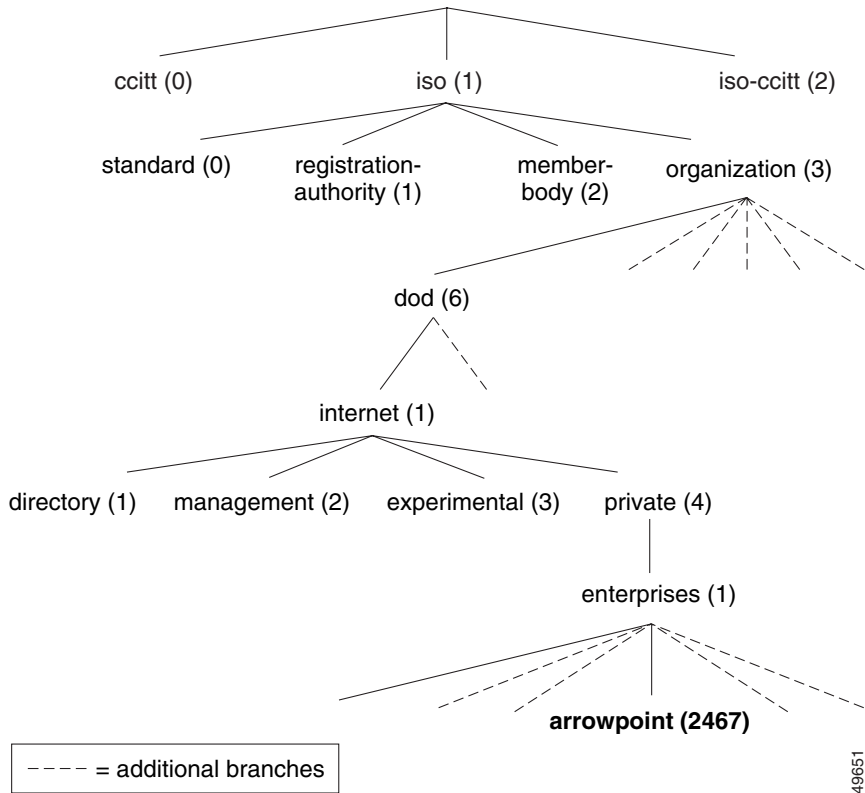
At the top of the tree is the broadest information about a network. Each branch and sub-branch of the tree gets progressively more specific, and the lowest branches of the tree contain the most specific MIB objects; the leaves contain the actual data. [Figure 12-2](#) shows an example of how the MIB tree objects become more specific as the tree expands.



Note

There are two versions of the MIB tree as defined by ISO: MIB-I and MIB-II. MIBII has more variables than MIB-I. Refer to the MIB-II standard in RFC 1213, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II."

Figure 12-2 Top of the MIB Tree



This section includes the following topics:

- [MIB Variables](#)
- [MIB Extensions \(Enterprise MIBs\)](#)
- [Updating MIB Files](#)

MIB Variables

There are two types of MIB variables:

- **Scalar** - Variables that define an object with a single representation. This means that the object describes a particular characteristic of the entire system. An example of a scalar variable is **SysDescr**, which provides a system-wide description of the CSS.
- **Tabular** - Variables that define an object with multiple representations. This means that the object can have different values, depending on the qualifier. For example, one tabular object could show bytes per interface, temperature per board, or hits per service.

As shown in [Figure 12-2](#), a number is associated with a MIB object name. This number is called the *object identifier* (or *object ID*), and it uniquely identifies the MIB object in the MIB tree. (The dotted lines represent other branches not relevant to this discussion.)

For example, note in [Figure 12-2](#) that the MIB object labeled *arrowpoint (2467)*, which contains the MIB objects specific to the CSS, can be labeled:

```
iso.organization.dod.internet.private.enterprises.arrowpoint  
or  
1.3.6.1.4.1.2467
```

MIB Extensions (Enterprise MIBs)

The MIB tree has a special branch set aside for specific vendors to build their own extensions; this special branch is called the *Enterprise MIB branch*. The CSS MIBs are included in the CSS GZIP file and are located in the CSS /mibs directory. The MIB files in this branch comprise the CSS Enterprise MIBs (the highlighted MIB identifier in [Figure 12-2](#)). The enterprise MIB files are categorized along functional boundaries.

For a list of MIB branches under the CSS Enterprise MIB, see the “[CSS MIBs](#)” section.

Updating MIB Files

We recommend that you update the CSS Enterprise MIBs after you upgrade the CSS software. CSS MIBs are included in the CSS GZIP file. During the software upgrade, the MIBs are loaded into the CSS /mibs directory.

To update the CSS MIBs on your management station after you upgrade the CSS:

1. Transfer the MIBs using FTP from the CSS MIBs (/v1 or /v2) directory to your management station.
2. Load the MIBs into the management application.

SNMP Communities

Each SNMP device or member is part of a *community*. An SNMP community determines the access rights for each SNMP device.

You supply a name to the community. After that, all SNMP devices that are assigned to that community as *members* have the same access rights. The access rights that the CSS supports are:

- read - Allows read-only access to the MIB tree for devices included in this community
- read-write - Allows both read and write access to the MIB tree for devices included in this community

Preparing to Configure SNMP on the CSS

Once you have set up your SNMP management application, you are ready to configure SNMP settings on the CSS. You can configure two basic areas of SNMP functionality on the CSS: SNMP functions and RMON functions.

**Note**

Refer to [Chapter 13, Configuring Remote Monitoring \(RMON\)](#) for information on configuring RMON.

To control SNMP access to the CSS, use the **no restrict snmp** and **restrict snmp** commands. Access through SNMP is enabled by default. The options for this global configuration mode command are:

- **no restrict snmp** - Enables SNMP access to the CSS (default setting)
- **restrict snmp** - Disables SNMP access to the CSS

Before you set up SNMP on your network consider the following items when planning your SNMP configuration:

- Decide which types of information the SNMP manager needs (if your application is using an SNMP manager). Choose the particular MIB objects that you want through the management software.
- Decide how many trap hosts you need. In some network configurations, you may want to have a primary trap host with one other workstation also receiving traps for redundancy. In a distributed or segmented network, you may want to have more trap hosts enabled. You can configure up to five trap hosts per SNMP agent; that is, one agent can report to a maximum of five hosts.
- Designate a management station or stations. The CSS is an agent in the SNMP network scheme. The agent is already embedded in the CSS when you boot up the device. All you need to do is configure the SNMP parameters on the CSS.

Defining the CSS as an SNMP Agent

This section describes how to define the CSS as an SNMP agent. It includes the following topics:

- [SNMP Agent Quick Start](#)
- [Configuring an SNMP Community](#)
- [Configuring an SNMP Contact](#)
- [Configuring an SNMP Location](#)
- [Configuring an SNMP Name](#)
- [Configuring SNMP Generic Traps](#)
- [Configuring an SNMP Trap-Host](#)
- [Configuring SNMP Source Traps](#)
- [Configuring SNMP Auth-Traps](#)
- [Configuring SNMP Enterprise Traps](#)
- [Configuring SNMP Reload-Enable](#)

SNMP Agent Quick Start

[Table 12-1](#) provides a quick overview of the steps required to configure the CSS as an SNMP agent. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following [Table 12-1](#).

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent

Task and Command Example

1. Define the SNMP community strings for each access type, read-only (for a GET action) or read-write (for a GET and SET action). This step is required for using SNMP on the CSS.

```
(config)# snmp community public read-only  
(config)# snmp community private read-write
```

2. (Optional) Provide the SNMP contact name.

```
(config)# snmp contact "fred n mandy"
```

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent (continued)

Task and Command Example

3. (Optional) Provide an SNMP contact location.

```
(config)# snmp location "Operations"
```

4. (Optional) Provide the SNMP device name.

```
(config)# snmp name "arrowpoint.com"
```

5. (Optional) Turn on generic traps.

```
(config)# snmp trap-type generic
```

6. Assign trap receivers and SNMP community (required if configuring SNMP traps). You can specify a maximum of five trap hosts. By default, all traps are disabled. The **snmp trap-host** IP address corresponds to the SNMP host configured to receive traps. The community information provided at the end of the **trap-host** command is included in the trap and can be used by the management station to filter incoming traps.

```
(config)# snmp trap-host 172.16.3.6 trap
```

```
(config)# snmp trap-host 172.16.8.4 trap
```

7. (Optional) Turn on authentication failure traps. An authentication failure occurs if an unauthorized SNMP manager sends an invalid or incorrect community name to an SNMP agent. If an authentication failure occurs, the agent sends an authentication trap to the trap host (or hosts depending on how many trap hosts are configured).

```
(config)# snmp auth-traps
```

8. (Optional) Enable global enterprise traps.

```
(config)# snmp trap-type enterprise
```

Enable a specific enterprise trap type. For example, you can set a trap to notify the trap host of failed login attempts. Login failure traps provide the username and source IP address of the person who failed to log in.

```
(config)# snmp trap-type enterprise login-failure
```

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent (continued)**Task and Command Example**

9. (Optional) Configure the trap host for reload enable ability. Reload enable allows a management station with the proper WRITE community privilege to reboot the CSS.

```
(config)# snmp reload-enable 100
```

10. (Optional) Configure special enterprise trap thresholds to notify the trap host of Denial of Service (DoS) attacks on your system. For example, you can set a trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination.

```
(config)# snmp trap-type enterprise dos-illegal-attack
trap-threshold 1
```

Configuring an SNMP Community

Use the **snmp community** command to set or modify SNMP community names and access privileges. You may specify as many community names as you wish.



Caution

You must define the community strings for each access type (read-only or read-write) before you use SNMP on the CSS. The CSS is inaccessible until you specify a read community string.

The syntax for this global configuration mode command is:

```
snmp community community_name [read-only|read-write]
```

The variables and options for this command are:

- *community_name* - The SNMP community name for this system. Enter an unquoted text string with no space and a maximum of 12 characters.
- **read-only** - Allows read-only access for this community.
- **read-write** - Allows read-write access for this community.

For example:

```
(config)# snmp community sqa read-write
```

To remove a community name, enter:

```
(config)# no snmp community sqa
```

Configuring an SNMP Contact

Use the **snmp contact** command to set or modify the contact name for the SNMP system. You can specify only one contact name. The syntax for this global configuration mode command is:

```
snmp contact "contact_name"
```

Enter the contact name as a quoted text string with a maximum of 255 characters including spaces. You can also include information on how to contact the person; for example, a phone number or e-mail address.

For example:

```
(config)# snmp contact "Fred N. Mandy"
```

To remove the specified SNMP contact name and reset it to the default of “Cisco Systems, Content Network Systems”, enter:

```
(config)# no snmp contact
```

Configuring an SNMP Location

Use the **snmp location** command to set or modify the SNMP system location. You can specify only one location. The syntax for this global configuration mode command is:

```
snmp location "location"
```

Enter the location as the physical location of the system. Enter a quoted text string with a maximum of 255 characters.

For example:

```
(config)# snmp location "sga_lab1"
```

To remove the specified SNMP system location and reset it to the default of “Customer Premises”, enter:

```
(config)# no snmp location
```

Configuring an SNMP Name

Use the **snmp name** command to set or modify the SNMP name for this system. You can specify only one name. The syntax for this global configuration mode command is:

```
snmp name "name"
```

Enter the SNMP name as the unique name assigned to a system by the administrator. Enter a quoted text string with a maximum of 255 characters. The standard name convention is the system’s fully qualified domain name (for example, sga@arrowpoint.com).

For example:

```
(config)# snmp name "sga@arrowpoint.com"
```

To remove the SNMP name for a system and reset it to the default of “Support”, enter:

```
(config)# no snmp name
```

Configuring SNMP Generic Traps

Use the **snmp trap-type generic** command to enable SNMP generic trap types. The generic SNMP traps consist of cold start, warm start, link down, and link up.



Note

For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the [“CSS SNMP Traps”](#) section.

For example:

```
(config)# snmp trap-type generic
```

To disable a generic trap, enter:

```
(config)# no snmp trap-type generic
```

Configuring an SNMP Trap-Host

Use the **snmp trap-host** command to set or modify the SNMP host to receive traps from a CSS. You can specify a maximum of five hosts. The syntax for this global configuration mode command is:

```
snmp trap-host ip_or_host community_name
```

The variables for this command are:

- *ip_or_host* - The IP address or host name of an SNMP host that has been configured to receive traps. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *community_name* - The community name to use when sending traps to the specified SNMP host. Enter an unquoted text string with no spaces and a maximum of 12 characters.

For example:

```
(config)# snmp trap-host 172.16.3.6 sqalab1
```

To remove a specified trap host, enter:

```
(config)# no snmp trap-host 172.16.3.6
```

Configuring SNMP Source Traps

Use the **snmp trap-source** command to set the source IP address in the traps generated by the CSS. The syntax of this global configuration mode command is:

```
snmp trap-source [egress-port|specified source_ip_address]
```

The options and variable for this command are:

- **egress-port** - Obtains the source IP address for the SNMP traps from the VLAN circuit IP address configured on the egress port used to send the trap. You do not need to enter an IP address because the address is determined dynamically by the CSS.
- **specified** *source_ip_address* - Allows you to enter the IP address to be used in the source IP field of the traps. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1)

For example:

```
(config)# snmp trap-source egress-port
```

To return SNMP source traps to the default of the management port IP address, enter:

```
(config)# no snmp trap-source
```

Configuring SNMP Auth-Traps

Use the **snmp auth-traps** command to enable reception of SNMP authentication traps. The CSS generates these traps when an SNMP management station attempts to access your system with invalid community names.



Note

For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the “[CSS SNMP Traps](#)” section.

For example:

```
(config)# snmp auth-traps
```

To disable reception of authentication traps, enter:

```
(config)# no snmp auth-traps
```

Configuring SNMP Enterprise Traps

Use the **snmp trap-type enterprise** command to enable SNMP enterprise trap types. You can enable the CSS to generate enterprise traps when:

- Denial of Service attack events occur
- A login fails
- A CSS service transitions state
- A power supply transitions state
- A module is inserted into a powered-on CSS chassis
- An Inter-Switch Communications (ISC) LifeTick failure message occurs

Use the **no** form of the **snmp trap-type enterprise** command to prevent the CSS from generating a trap when a specific condition occurs.

For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the “[CSS SNMP Traps](#)” section. For information on configuring Denial of Service enterprise traps, see the “[Configuring Denial of Service \(DoS\)](#)” section.

The syntax for this global configuration mode command is:

```
snmp trap-type enterprise {dos_attack_type {trap-threshold  
threshold_value}}|chmgr-module-transition|chmgr-ps-transition  
|isc-lifetick-failure|login-failure|reload|redundancy-transition  
|service-transition}
```

The options for this command are as follows:

- **snmp trap-type enterprise** - Enables enterprise traps. You must enable enterprise traps before you configure an enterprise trap option.

- *dos_attack_type* - (Optional) Generates SNMP enterprise traps when a Denial of Service (DoS) attack event occurs. One trap is generated each second when the number of attacks during that second exceeds the threshold for the configured DoS attack type. See the “[Configuring Denial of Service \(DoS\)](#)” section for details.
- **trap-threshold** *threshold_value* - (Optional) Overrides a default trap threshold. For the *threshold_value*, enter a number from 1 to 65535. See the “[Configuring Denial of Service \(DoS\)](#)” section for details.
- **chmgr-module-transition** - Generates SNMP enterprise traps if a module (for example, SCM or SSL) is inserted into or removed from a powered-on CSS 11503 or CSS 11506.
- **chmgr-ps-transition** - Generates SNMP enterprise traps when the CSS 11503 or CSS 11506 power supply changes state (powered off or on, or removed from the CSS).
- **isc-lifetick-failure** - Generates SNMP enterprise traps when an ISC LifeTick failure message occurs on a CSS. A LifeTick message occurs four times a second between ports in an Adaptive Session Redundancy (ASR) configuration. If a port does not receive a LifeTick message within one second from its corresponding port due to a software or hardware failure, an ISC LifeTick failure message occurs.
- **login-failure** - Generates SNMP enterprise traps when a CSS login failure occurs. The CSS also generates an alert-level log message.
- **reload** - Generates SNMP enterprise traps when a CSS reboot occurs. The CSS also generates a trap when a reboot is initiated directly through SNMP.
- **redundancy-transition** - Generates SNMP enterprise traps when the CSS redundancy transitions state.
- **service-transition** - Generates SNMP enterprise traps when a CSS service transitions state. A trap is generated when a service fails and when a failed service resumes proper operation.

To enable an SNMP enterprise trap when a CSS login failure occurs, enter:

```
(config)# snmp trap-type enterprise login-failure
```

To disable all enterprise traps, enter:

```
(config)# no snmp trap-type enterprise
```

To disable a specific enabled enterprise trap, use the **no** form of the **snmp trap-type enterprise** command. To prevent the CSS from generating traps when a power supply fails, enter:

```
(config)# no snmp trap-type enterprise chmgr-ps-transition
```

Configuring SNMP Reload-Enable

Use the **snmp reload-enable** command to reboot the CSS using SNMP. The syntax and options for this global configuration mode command are:

- **snmp reload-enable** - Allows any SNMP write to the `apSnmExtReloadSet` object to force a CSS reboot. The reload object, `apSnmExtReloadSet`, is located at 1.3.6.1.4.1.2467.1.22.7. You can find this object in the CSS Enterprise MIB, `snmpext.mib`.
- **snmp reload-enable reload_value** - Allows an SNMP write equal to the `reload_value` to force a CSS reboot.

Enter the `reload_value` as the object used to control `apSnmExtReloadSet`, providing the SNMP-based reboot. When the object is set to 0, an SNMP reboot is not allowed. When the object is set from 1 to 232, a reboot may be caused with any write value to `apSnmExtReloadSet`. For security purposes, this object always returns 0 when read.

For example:

```
(config)# snmp reload-enable
```

To prevent users from rebooting the CSS using SNMP (default behavior), enter:

```
(config)# no snmp reload-enable
```

Configuring Denial of Service (DoS)

You can configure special enterprise traps to notify the trap host of Denial of Service (DoS) attacks on your system. You can also use the CLI to display detailed information about DoS attacks and reset the DoS statistics for your CSS to zero.

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, see the “[Configuring SNMP Enterprise Traps](#)” section.

This section includes the following topics:

- [DoS Quick Start](#)
- [Defining a DoS SNMP Trap-Type](#)
- [Displaying DoS Configurations](#)

DoS Quick Start

[Table 12-2](#) provides a quick overview of the steps required to configure the CSS as an SNMP agent. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following [Table 12-2](#).

Table 12-2 Denial of Service Configuration Quick Start

Task and Command Example

1. Set the trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination.

```
(config)# snmp trap-type enterprise dos-illegal-attack
trap-threshold 1
```

2. Set the trap threshold to notify the trap host of DoS LAND attacks.

```
(config)# snmp trap-type enterprise dos-land-attack
trap-threshold 1
```

3. Set the trap threshold to notify the trap host of DoS smurf attacks.

```
(config)# snmp trap-type enterprise dos-smurf-attack
trap-threshold 1
```

Table 12-2 Denial of Service Configuration Quick Start (continued)

Task and Command Example
4. Set the trap threshold to notify the trap host of DoS SYN attacks. <pre>(config)# snmp trap-type enterprise dos-syn-attack trap-threshold 10</pre>
5. Display information about DoS attacks. <pre>(config)# show dos summary (config)# show dos</pre>
6. Reset the DoS statistics for a CSS to zero, as required. <pre>(config)# zero dos statistics</pre>

Defining a DoS SNMP Trap-Type

Use the **snmp trap-type enterprise** command to enable the CSS to generate SNMP enterprise traps when a DoS attack event occurs. One trap is generated each second when the number of attacks during that second exceeds the threshold for the configured DoS attack type. For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the “[CSS SNMP Traps](#)” section.

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, see the “[Configuring SNMP Enterprise Traps](#)” section.

The syntax for this global configuration mode command is:

```
snmp trap-type enterprise dos_attack_type {trap-threshold  
threshold_value}
```

The *dos_attack_type* variable is the type of DoS attack event to trap. The options include:

- **dos-illegal-attack** - Generates traps for illegal addresses, either source or destination. Illegal addresses are loopback source addresses, broadcast source addresses, loopback destination addresses, multicast source addresses, or source addresses that you own. The default trap threshold for this type of attack is 1 per second.
- **dos-land-attack** - Generates traps for packets that have identical source and destination addresses. The default trap threshold for this type of attack is 1 per second.
- **dos-smurf-attack** - Generates traps when the number of pings with a broadcast destination address exceeds the threshold value. The default trap threshold for this type of attack is 1 per second.
- **dos-syn-attack** - Generates traps when the number of TCP connections that are initiated by a source, but not followed with an acknowledgment (ACK) frame to complete the 3-way TCP handshake, exceeds the threshold value. The default trap threshold for this type of attack is 10 per second.

Use the **trap-threshold** option to override a default trap threshold. For the *threshold_value*, enter a number from 1 to 65535.

For example, to enable the CSS to generate traps for packets that have identical source and destination addresses, enter:

```
(config)# snmp trap-type enterprise dos-land-attack
```

To prevent the CSS from generating DoS attack event traps, enter:

```
(config)# no snmp trap-type enterprise dos_attack_type
```

Displaying DoS Configurations

Use the **show dos** command to display detailed information about DoS attacks on each CSS Session Processor. The **show dos** command displays the following information:

- The total number of attacks since booting the CSS
- The types of attacks and the maximum number of these attacks per second
- The first and last occurrence of an attack
- The source and destination IP addresses

A CSS can display a maximum of 50 of the most recent attack events for each SP. For example:

- A CSS 11501 with one SP can display a maximum of 50 events.
- A CSS 11503 with a maximum of three SPs can display a maximum of 150 events.
- A CSS 11506 with a maximum of six SPs can display a maximum of 300 events.

If multiple attacks occur with the same DoS type and source and destination address, an attempt is made to merge them as one event. This merging of events reduces the number of displayed events.

Use the **show dos summary** command to display a summary of information about DoS attacks.

For example:

```
(config)# show dos summary
```

[Table 12-3](#) describes the fields in the **show dos** command output.

Table 12-3 Field Descriptions for the show dos Command

Field	Description
Total Attacks	<p>The total number of DoS attacks detected since the CSS was booted. The type of attacks that are listed along with their number of occurrences are:</p> <ul style="list-style-type: none"> • SYN Attacks - TCP connections that are initiated by a source but are not followed with an ACK frame to complete the three way TCP handshake • LAND Attacks - Packets that have identical source and destination addresses • Zero Port Attacks - Frames that contain source or destination TCP or UDP ports equal to zero <p>Note Older SmartBits software may send frames containing source or destination ports equal to zero. The CSS logs them as DoS attacks and drops these frames.</p> <ul style="list-style-type: none"> • Illegal Src Attacks - Illegal source addresses • Illegal Dst Attacks - Illegal destination addresses • Smurf Attacks - Pings with a broadcast destination address
First Attack Detected	The first time a DoS attack was detected.
Last Attack Detected	The last time a DoS attack was detected.

Table 12-3 Field Descriptions for the show dos Command (continued)

Field	Description
Maximum per second	<p>The maximum number of events per second. Use the maximum events per second information to set SNMP trap threshold values. The maximum number of events per second is the maximum for each SP.</p> <ul style="list-style-type: none"> For a CSS 11506, which may have up to six SPs, the maximum rate per second may be as high as six times the value appearing in this field. For a CSS 11503, which may have up to three SPs, the maximum rate per second may be as high as three times the value appearing in this field.
DoS Attack Event	Details for each detected attack event, up to a maximum of 50 events per SP.
First Attack	The first time the attack event occurred.
Last Attack	The last time the attack event occurred.
Source/Destination Address	The source and destination addresses for the attack event.
Event Type	The type of event.
Total Attacks	The total number of attack occurrences for the event.

Displaying the SNMP Configuration

After you configure SNMP, display the SNMP configuration. For example:

```
(config)# show running-config global
```

Refer to [Chapter 3, Managing the CSS Software](#) for details on the **show running-config** command and its output.

Managing SNMP on the CSS

This section describes the activities that you need to perform to manage SNMP on the CSS. This section includes the following topics:

- [Enabling SNMP Manager Access to the CSS](#)
- [Using the CSS to Look Up MIB Objects](#)
- [Reading Logs](#)
- [Setting RMON Alarms](#)

Enabling SNMP Manager Access to the CSS

By default, the CSS enables SNMP access to its command base. You must first create community strings using the **snmp community** command before you can use SNMP in the CSS. See the “[Configuring an SNMP Community](#)” section for details.



Note

SNMP is not a secure network environment. Do not use SNMP by itself to provide security for your network.

Using the CSS to Look Up MIB Objects

To look up a MIB object, including the variables that make up the object, perform the following steps:

1. Access global configuration mode by entering:

```
# config
```

2. Access rmon-alarm mode by entering:

```
(config)# rmon-alarm index_number
```

where *index_number* is the RMON alarm index. The RMON alarm index identifies the alarm to the CSS. Refer to [Chapter 13, Configuring Remote Monitoring \(RMON\)](#) for information on RMON.

3. Display the MIB object by entering:

```
(config-rmonalarm[1])# lookup object
```

where *object* is the name of the MIB object.

You can look up a specific object, or you can use the question mark (?) character as a wildcard to help you complete your request.

For example, suppose you want to look up a MIB object but you are not sure of its exact name. You already know that the MIB you want is part of the `apFlowMgrExt` group of objects. In this case, specify the **lookup** command with the question mark (?) character, as shown.

```
(config-rmonalarm[1])# lookup apFlowMgrExt?
```

```
apFlowMgrExtDoSAttackEventType  
apFlowMgrExtDoSAttackEventCount  
apFlowMgrExtDoSAttackIndex  
apFlowMgrExtDosTotalSmurfAttacks  
apFlowMgrExtDosTotalIllegalSourceAttacks  
apFlowMgrExtDosTotalZeroPortAttacks  
apFlowMgrExtDosTotalLandAttacks  
apFlowMgrExtDosTotalSynAttacks  
apFlowMgrExtDosTotalAttacks  
apFlowMgrExtIdleTimer  
apFlowMgrExtPortIdleValue  
apFlowMgrExtPortIdle  
apFlowMgrExtReserveCleanTimer  
apFlowMgrExtPermanentPort4  
apFlowMgrExtPermanentPort3
```

```

apFlowMgrExtPermanentPort2
apFlowMgrExtPermanentPort1
apFlowMgrExtFlowTraceDuration
apFlowMgrExtFlowTraceMaxFileSize
apFlowMgrExtFlowTraceState

```

The previous example shows that using the question mark (?) character as a wildcard returns information about the apFlowMgrExt MIB object. You can also enter the **lookup** command on the exact MIB you want and view its description without using the question mark (?) character. For example:

```

(config-rmonalarm[1])# lookup apFlowMgrExtDOSAttackEventCount

ASN Name:          apFlowMgrExtDOSAttackEventCount
MIB:              flowmgrext
Object Identifier: 1.3.6.1.4.1.2467.1.36.27.1.6
Argument Type:    Integer
Range:            0-4294967295
Description:
    This is the number of times this DoS attack had occurred.

```

You can also display a list of all the Enterprise MIBs by using the **lookup** command without any MIB object names, as shown in the following example:

```

(config-rmonalarm[1])# lookup ?

```

The **lookup** command omits MIB objects of type *string* and *MAC address*.

Useful MIB Information

Table 12-4 lists some of the MIB groups that provide useful information about the CSS.

Table 12-4 CSS MIB Information

MIB Name	Description
RFC 1398	Ethernet statistics
RFC 1493	Bridge information
RFC 1757	RMON statistics
svcExt.mib	Service variables (including TCP connections)
cntExt.mib	Content rule variables (including frame statistics)

Table 12-4 CSS MIB Information (continued)

MIB Name	Description
ownExt.mib	Owner statistics (including frame and bytes counts)
cntsvcExt.mib	Services per content rule statistics (including frames, bytes, hits)
chassis MgrExt	Provides useful information about the CSS chassis and it allows you to correlate the slot number and port number to the ifIndex number

Reading Logs

The traplog file contains all of the traps, both generic and enterprise, that have occurred. The network device writes to the traplog file about whether or not the SNMP trap configuration is enabled.

Use the **show log** command to show the trap log since the last CSS reboot. For example:

```
# show log traplog
```

By default, the following events generate level critical-2 messages:

- Link Up
- Link Down
- Cold Start
- Warm Start
- Service Down
- Service Suspended

All other SNMP traps generate level notice-5 messages by default.

Setting RMON Alarms

An RMON alarm allows you to monitor a MIB object for a desired transitory state. Refer to [Chapter 13, Configuring Remote Monitoring \(RMON\)](#) for information about commands available in the RMON alarm mode.

CSS SNMP Traps

Table 12-5 lists the SNMP traps supported by the CSS.

Table 12-5 *SNMP Traps*

Name/MIB	Enterprise Object ID (OID)	Generic	Specific	Parameters
coldStart	<sysObjectID>	0	0	_____
warmStart	<sysObjectID>	1	0	_____
linkDown	<sysObjectID>	2	0	ifIndex 1.3.6.1.2.1.2.2.1.1
linkUp	<sysObjectID>	3	0	ifIndex 1.3.6.1.2.1.2.2.1.1
authenticationFailure	<sysObjectID>	4	0	_____
egpNeighborLoss	<sysObjectID>	5	0	_____
apFlowMgrExtDosSynTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	1	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventInterval Count 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apFlowMgrExtDosLandTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	2	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventInterval Count 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apFlowMgrExtDosIllegalTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	3	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventInterval Count 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6

Table 12-5 SNMP Traps (continued)

Name/MIB	Enterprise Object ID (OID)	Generic	Specific	Parameters
apFlowMgrExtDosSmurfTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	5	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventInterval Count 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apIpv4RedundancyTrap (apIpv4.mib)	1.3.6.1.4.1.2467.1.9.1	6	1	apIpv4TrapEventText 1.3.6.1.4.1.2467.1.9.34.0 apIpv4RedundancyState 1.3.6.1.4.1.2467.1.9.19.0 apIpv4RedundancyIf 1.3.6.1.4.1.2467.1.9.20.0 apIpv4RedundancyMaster 1.3.6.1.4.1.2467.1.9.21.0
apSnmExtReloadTrap (snmpExt.mib)	1.3.6.1.4.1.2467.1.22	6	1	apSnmExtTrapEventText 1.3.6.1.4.1.2467.1.22.27.0
apSvcTransitionTrap (svcExt.mib)	1.3.6.1.4.1.2467.1.15	6	1	apSvcTrapEventText 1.3.6.1.4.1.2467.1.15.10.0
apTermSessLoginFailureTrap (terminalMgmt.mib)	1.3.6.1.4.1.2467.1.11	6	1	apTermSessLoginFailureInfo 1.3.6.1.4.1.2467.1.11.3.0
apChassisMgrExtPsTrap (chassisMgrExt.mib)	1.3.6.1.4.1.2467.1.34	6	1	apChassisMgrExtTrapPsEventText 1.3.6.1.4.1.2467.1.34.24.0
apChassisMgrModuleTrap (chassisMgrExt.mib)	1.3.6.1.4.1.2467.1.34	6	2	apChassisMgrExtTrapModuleEventText 1.3.6.1.4.1.2467.1.34.25.0
apEnetISCLifeticTrap (enetExt.mib)	1.3.6.1.4.1.2467.1.39	6	1	apEnetISCLifeticEventText 1.3.6.1.4.1.2467.1.39.8.0

CSS MIBs

Table 12-6 describes the CSS MIB objects directly under the CSS Enterprise MIB (Object Identifier 1.3.6.1.4.1.2467). The MIBs listed in this table are a representation of the CSS content-specific MIB objects. To find out how you can look up object information, see the “Using the CSS to Look Up MIB Objects” section.

Table 12-6 MIB Branches Under the CSS Enterprise MIB

MIB Filename	MIB Module Description	Related CLI Commands
aclExt.mib (OID 1.3.6.1.4.1.2467.1.23)	The CSS access control list (ACL) clause table	(config-acl)# ?
ap64Stats.mib (OID 1.3.6.1.4.1.2467.1.44)	The 64-bit statistical aggregation of RMON (RFC1757), MIB-II (RFC 1213) and EtherErrors (RFC 1398)	# show rmon ? # show mibii ? # show ether-errors ?
apent.mib (OID 1.3.6.1.4.1.2467.1)	CSS Enterprise MIB branch hierarchy	-----
apIpv4.mib (OID 1.3.6.1.4.1.2467.1.9.1)	MIB support for IPv4 global information, box-to-box redundancy	(config)# ip ?
apIpv4Arp.mib (OID 1.3.6.1.4.1.2467.1.9.4)	MIB support for IPv4 ARP	(config)# arp ?
apIpv4Dns.mib (OID 1.3.6.1.4.1.2467.1.9.7)	MIB support for IPv4 DNS resolver configuration	(config)# dns ?
apIpv4Host.mib (OID 1.3.6.1.4.1.2467.1.9.6)	MIB support for IPv4 host table	(config)# host ?
apIpv4Interface.mib (OID 1.3.6.1.4.1.2467.1.9.2)	MIB support for IPv4 interfaces, box-to-box redundancy	(config-ip)# ?
apIpv4Ospf.mib (OID 1.3.6.1.4.1.2467.1.9.3.2)	MIB support for the Open Shortest Path First (OSPF) protocol	(config)# ospf ?
apIpv4Redundancy.mib (OID 1.3.6.1.4.1.2467.1.9.8)	MIB support for IPv4 redundancy	(config-ip)# redundancy ?

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
apIpv4Rip.mib (OID 1.3.6.1.4.1.2467.1.9.3.1)	MIB support for the Routing Information Protocol (RIP)	(config-ip)# rip ?
apIpv4Sntp.mib (OID 1.3.6.1.4.1.2467.1.9.9)	MIB support for the Simple Network Time Protocol (SNTP)	(config)# sntp ?
apIpv4StaticRoutes.mib (OID 1.3.6.1.4.1.2467.1.9.5)	MIB support for IPv4 static routes	(config)# ip route ?
appExt.mib (OID 1.3.6.1.4.1.2467.1.32)	MIB support for Application Peering Protocol (APP) configurations	(config)# app ?
boomClientExt.mib (OID 1.3.6.1.4.1.2467.1.62)	Configuration and monitoring of Content Routing Agent (CRA) parameters	(config)# dns-boomerang client ?
bootExt.mib (OID 1.3.6.1.4.1.2467.1.31)	MIB support for system boot administration	(config-boot)# ?
bridgeExt.mib (OID 1.3.6.1.4.1.2467.1.14)	Configuration and monitoring of bridge-related parameters	(config)# bridge ?
cappUdpExt.mib (OID 1.3.6.1.4.1.2467.1.52)	Application Peering Protocol-User Datagram Protocol (APP-UDP) global statistical information and security configuration settings	(config)# app-udp ?
cctExt.mib (OID 1.3.6.1.4.1.2467.1.29)	CSS circuit information, box-to-box redundancy	(config)# circuit ?
chassisMgrExt.mib (OID 1.3.6.1.4.1.2467.1.34)	MIB for the CSS chassis manager	# show chassis ?
cntdnsExt.mib (OID 1.3.6.1.4.1.2467.1.41)	Content rule Domain Name Service (DNS) statistics	(config)# dns hotlist ?
cntExt.mib (OID 1.3.6.1.4.1.2467.1.16)	Content rule table	(config-owner-content)# ?
cnthotExt.mib (OID 1.3.6.1.4.1.2467.1.35)	Content rule hot list	(config-owner-content)# hotlist ?

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
cntsvcExt.mib (OID 1.3.6.1.4.1.2467.1.18)	Monitoring of services attached to content rules	(config-owner-content)# add service ? (config-owner-content)# remove service ?
csaExt.mib (OID 1.3.6.1.4.1.2467.1.59)	Configuration and monitoring of Client Side Accelerator (CSA) parameters on a CSS	(config)# dns-server ?
dfpExt.mib (OID 1.3.6.1.4.1.2467.1.65)	MIB support for Dynamic Feedback Protocol (DFP) statistics and configuration	(config)# dfp ?
dnshotExt.mib (OID 1.3.6.1.4.1.2467.1.48)	DNS hot list	(config)# domain hotlist ?
dnsServerExt.mib (OID 1.3.6.1.4.1.2467.1.40)	MIB support for DNS server	(config)# dns-server ?
domainCacheExt.mib (OID 1.3.6.1.4.1.2467.1.60)	Configuration management for the domain cache on the CSA in the CSS	(config)# dns-server domain-cache ?
dqlExt.mib (OID 1.3.6.1.4.1.2467.1.51)	Domain Qualifier Lists (DQLs)	(config-dql [name])# ?
enetExt.mib (OID 1.3.6.1.4.1.2467.1.39)	Configuration of the PHY state for Ethernet ports	(config-interface)# phy ?
eqlExt.mib (OID 1.3.6.1.4.1.2467.1.42)	Extension Qualifier Lists (EQLs)	(config-eql [name])#
fileExt.mib (OID 1.3.6.1.4.1.2467.1.61)	File extensions to support network management movement to/from the CSS, and to examine and modify the existing file structure	-----
flowMgrExt.mib (OID 1.3.6.1.4.1.2467.1.36)	MIB for the flow manager module	(config)# flow ?
ftpExt.mib (OID 1.3.6.1.4.1.2467.1.30)	MIB support for File Transfer Protocol (FTP) transfer administration records	(config)# ftp-record ?

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
grpExt.mib (OID 1.3.6.1.4.1.2467.1.17)	Configuration of all group-related parameters	(config-group)# ?
grpsvcExt.mib (OID 1.3.6.1.4.1.2467.1.19)	Groups attached to services	(config-group)# add service ? (config-group)# remove service ?
httpExt.mib (OID 1.3.6.1.4.1.2467.1.47)	MIB support for HTTP transfer administration records	-----
kalExt.mib (OID 1.3.6.1.4.1.2467.1.46)	Configuration of keepalive mode	(config-keepalive)# ?
logExt.mib (OID 1.3.6.1.4.1.2467.1.20)	CSS logging functionality	(config)# logging ?
nqlExt.mib (OID 1.3.6.1.4.1.2467.1.50)	Describes the CSS network qualifier lists (NQLs)	(config-nql [name])# ?
ownExt.mib (OID 1.3.6.1.4.1.2467.1.25)	Web host owner information	(config-owner)# ?
plucExt.mib (OID 1.3.6.1.4.1.2467.1.56)	Proximity Lookup Client functionality	(config)# proximity cache ?
probeRttExt.mib (OID 1.3.6.1.4.1.2467.1.55)	Tiered Proximity Service RTT Probe Module functionality	(config)# proximity probe rtt ?
proxDbExt.mib (OID 1.3.6.1.4.1.2467.1.54)	Tiered Proximity Database (PDB) functionality; contains all configuration, statistic, and metric objects	(config)# proximity db ?
publishExt.mib (OID 1.3.6.1.4.1.2467.1.57)	Publisher and subscriber services	(config-service)# publisher ?
qosExt.mib (OID 1.3.6.1.4.1.2467.1.28)	CSS MIB module quality of service (QoS) class definitions (the QoS class of this known piece of content)	-----
radiusClientExt.mib (OID 1.3.6.1.4.1.2467.1.12)	CSS extensions to the client side of the Remote Access Dial-in User Service (RADIUS) authentication protocol	(config)# radius-server ?

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
schedExt.mib (OID 1.3.6.1.4.1.2467.1.45)	MIB support for CLI command scheduler records	(config)# cmd-scheduler ?
securityMgrExt.mib (OID 1.3.6.1.4.1.2467.1.13)	CSS MIB objects for the network security manager	(config)# username ?
snmpExt.mib (OID 1.3.6.1.4.1.2467.1.22)	SNMP traps and communities	(config)# snmp ?
sshdExt.mib (OID 1.3.6.1.4.1.2467.1.43)	MIB support for the Secure Shell Daemon server (SSHD)	(config)# sshd ?
sslExt.mib (OID 1.3.6.1.4.1.2467.1.63)	MIB support for Secure Sockets Layer (SSL) file associations for SSL certificates and keys for the SSL Acceleration Module	(config)# ssl cert ? (config)# ssl rsa key ? (config)# ssl da key ? (config)# ssl dh parm ?
sslExt.mib (OID 1.3.6.1.4.1.2467.1.64)	MIB support for SSL proxy list elements and cipher suite objects for the SSL Acceleration Module	(ssl-proxy-list[name])# element ?
subscribeExt.mib (OID 1.3.6.1.4.1.2467.1.58)	CSS Enterprise subscriber	(config-service)# subscriber ?
svcExt.mib (OID 1.3.6.1.4.1.2467.1.15)	Configuration and monitoring of all service-related parameters	(config-service)# ?
tacacsExt.mib (OID 1.3.6.1.4.1.2467.1.66)	CSS extensions to the client side of the Terminal Access Controller Access Control System (TACACS+) authentication protocol	(config)# tacacs-server ?
tagExt.mib (OID 1.3.6.1.4.1.2467.1.53)	Content tag lists	(config)# header-field-group ?
terminalMgmt.mib (OID 1.3.6.1.4.1.2467.1.11)	MIB support for terminal options	# terminal ? # restrict ?
urqlExt.mib (OID 1.3.6.1.4.1.2467.1.49)	Uniform resource locator qualifier lists (URQL)	(config-urql [name])# ?

Where to Go Next

[Chapter 13, Configuring Remote Monitoring \(RMON\)](#), describes how to describe how to configure RMON on the CSS.

■ Where to Go Next