This chapter describes how to configure the the ML1000-2, ML100T-12, ML100X-8, and ML-MR-10 cards for operating with Simple Network Management Protocol (SNMP).

Note: For complete syntax and usage information for the commands used in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.

This chapter consists of these sections:

- Understanding SNMP
- Configuring SNMP
- Displaying SNMP Status

Contents

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a MIB. The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. To configure SNMP, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value in an agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

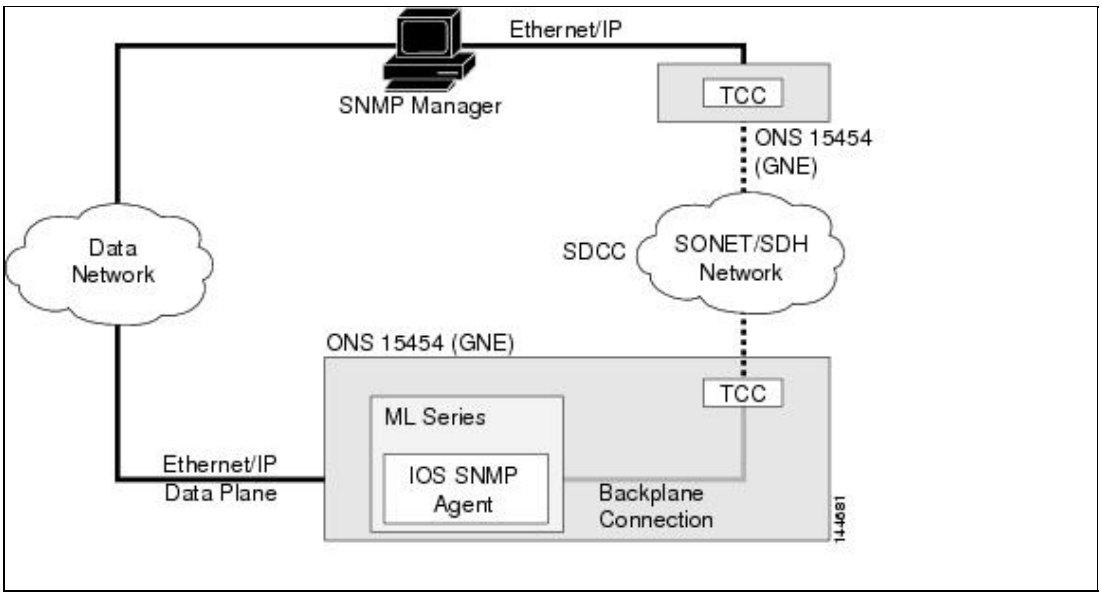
An agent can send unsolicited traps to the manager. Traps are messages that alert the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events.

This section includes information about these topics:

SNMP on the ML-Series Card

SNMP operates in two different ways on the ONS 15454 SONET/SDH ML-Series card. One way is to communicate directly. This is also how SNMP operates on a small Catalyst switch, using direct communication, Cisco IOS, and the data plane. An SNMP agent interacting with an ML-Series card can also communicate through the ONS 15454 SONET/SDH and the SONET network. Both ways are shown in Figure 24-1.

Figure 24-1: SNMP on the ML-Series Card Example



When the ONS 15454 SONET/SDH node relays the ML-Series card SNMP communication, the node uses a proxy agent to accept, validate, and forward get, getNext, and set requests to the ML-Series card. These ML-Series card requests contain the slot identification of the ML-Series card cards to distinguish the request from a general SNMP request for the ONS 15454 SONET/SDH node. The responses from the ML-Series card are then relayed by the ONS 15454 SONET/SDH node to the requesting SNMP agents.

SNMP access is useful for collecting Cisco IOS data plane events, alarms, and statistics for the ML-Series card. All SNMP events and traps defined on the ML-Series card are reported to the TCC2/TCC2P card SNMP agent by default. If the TCC2/TCC2P card SNMP agent is active, these events are sent to the defined SNMP server.

SNMP Versions

Both the ML-Series card and the ONS 15454 SONET/SDH nodes support SNMP Version 1 (SNMPv1) and SNMP Version 2c (SNMPv2c), defined as:

- SNMPv1-The Simple Network Management Protocol, a full Internet standard, defined in RFC 1157.
- SNMPv2c replaces the party-based administrative and security framework of SNMPv2 classic with the community-string-based administrative framework of SNMPv2c while retaining the bulk retrieval and improved error handling of SNMPv2classic. The bulk retrieval mechanism retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2c improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2c report the error type.

SNMPv1 and SNMpv2c have the same security models and levels:

- Level-noAuthNoPriv
- Authentication-community string
- Encryption-none
- Result-Uses a community string match for authentication.

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, and SNMPv2c protocols.

Figure 24-1: SNMP on the ML-Series Card Example

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in Table 24-1.

Table 24-1: SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table.[1]
get-bulk-request[2]	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, or set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

1. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table. 2. The **get-bulk-request** command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- Get a MIB variable-The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Set a MIB variable-The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the ML-Series card, the community string definitions on the NMS must match at least one of the three community string definitions on the ML-Series card.

A community string can have one of these attributes:

- Read-only (RO)-Gives read access to authorized management stations to all objects in the MIB except the community strings, but does not allow write access
- Read-write (RW)-Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings
- Read-write-all-Gives read and write access to authorized management stations to all objects in the MIB, including the community strings

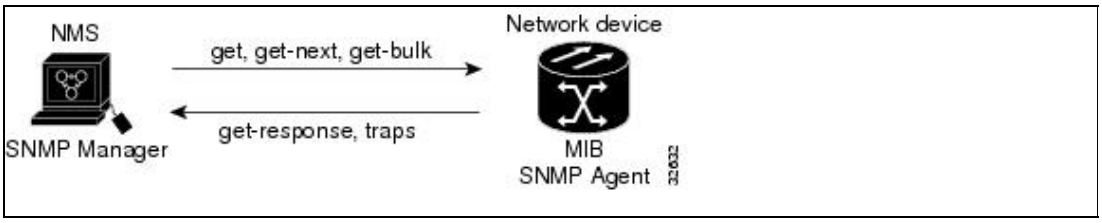
Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks software uses the ML-Series card MIB variables to set device variables and to poll devices on the network for specific

information. The results of a poll can be displayed as a graph and analyzed to troubleshoot problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in Figure 24-2, the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in get-request, get-next-request, and set-request format.

Figure 24-2: SNMP Network



Supported MIBs

The complete list of supported MIBs for the ML-Series card is found in the MIBs README.txt file on the ONS Software CD for your release. This software CD also includes the needed MIB modules and information on loading MIBs.

You can also locate and download MIBs for Cisco platforms, Cisco IOS releases, and feature sets, using the Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

SNMP Traps Supported on ML-MR-10 Card

The following traps are supported only on the ML-MR-10 card.

Table 24-2: Traps Supported on ML-MR-10 Card

Operation	Description
config traps	snmp-server enable traps conf
config-copy traps	snmp-server enable traps config-copy
cpu traps	snmp-server enable traps cpu
entity traps	snmp-server enable trap s entity
snmp linkup traps	snmp-server enable traps snmp linkup
snmp linkdown traps	snmp-server enable traps snmp linkdown

SNMP Notifications

SNMP allows the ML-Series card to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or as inform requests. In command syntax, unless there is an option in the command to select either traps or inform requests, the keyword *traps* refers to either traps or inform requests, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or inform requests.

Note: SNMPv1 does not support inform requests.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, so the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be re-sent, inform requests are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the ML-Series card and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be re-sent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the ML-Series card is a concern and notification is not required, use traps.

Configuring SNMP

This section describes how to configure SNMP on your ML-Series card. It contains this configuration information:

Default SNMP Configuration

Table 24-3 shows the default SNMP configuration.

Table 24-3: Default SNMP Configuration

Feature	Default Setting
SNMP agent	Enabled
SNMP community strings	Read-Only: Public Read-Write: Private Read-Write-all: Secret
SNMP trap receiver	None configured
SNMP traps	None enabled except the trap for TCP connections (**tty**)
SNMP version	If no **version** keyword is present, the default is Version 1.
SNMP notification type	If no type is specified, all notifications are sent.

SNMP Configuration Guidelines

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command autogenerates a notify view for the user and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group. For information about when you should configure notify views, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.
- An SNMP *group* is a table that maps SNMP users to SNMP views.
- An SNMP *user* is a member of an SNMP group.
- An SNMP *host* is the recipient of an SNMP trap operation.
- An SNMP *engine ID* is a name for the local or remote SNMP engine.

Disabling the SNMP Agent

Beginning in privileged EXEC mode, follow these steps to disable the SNMP agent:

Step	Command	Purpose
1	**router# configure terminal**	Enter global configuration mode.
2	**router (config)# no snmp-server**	Disable the SNMP agent operation.
3	router (config)# end	Return to privileged EXEC mode.

The **no snmp-server** global configuration command disables all running versions on the device. No specific Cisco IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables all versions of SNMP.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the ML-Series card card. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

Beginning in privileged EXEC mode, follow these steps to configure a community string on the ML-Series card:

Step	Command	Purpose
1	router# **configure terminal**	Enter global configuration mode.
2	router (config)# **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]	Configure the community string. • For *string*, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings of any length. • (Optional) For **view** *view-name*, specify the view record accessible to the community. • (Optional) Specify either read-only (**ro**) if you want authorized management stations to retrieve MIB objects, or specify read-write (**rw**) if you want authorized management stations to retrieve and modify MIB objects. By default,

		the community string permits read-only access to all objects. • (Optional) For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
3	`router (config)# access-list access-list-number {deny | permit} source [source-wildcard]`	(Optional) If you specified an IP standard access list number in Step 2, then create the list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the SNMP manager that are permitted to use the community string to gain access to the agent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
4	**`router (config)# end`**	Return to privileged EXEC mode.

Note: To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

To remove a specific community string, use the **no snmp-server community** *string* global configuration command.

This example shows how to assign the string comaccess to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the ML-Series card SNMP agent:

```
ML_Series(config)# snmp-server community comaccess ro 4
```

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the ML-Series card. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

Beginning in privileged EXEC mode, follow these steps to configure SNMP on the ML-Series card:

Step	Command	Purpose
1	`router# configure terminal`	Enter global configuration mode.
2	`router (config)# snmp-server engineID {local engineid-string | remote ip-address [udp-port port-number]}`	Configure a name for either the local or remote copy of SNMP. • The *engineid-string* is a 24-character ID string with the name of the copy of

		SNMP. • If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional UDP port on the remote device. The UDP port default is 162.
3	`router (config)# snmp-server group` *groupname* {**v1** | **v2c** [**auth** | **noauth** | **priv**]} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]	Configure a new SNMP group on the remote device. • For *groupname,* specify the name of the group. • Specify a security model: ♦ **v1** is the less secure model. ♦ **v2c** is the more secure model. It allows transmission of inform requests and integers that are twice the normal width. **Note:** The **priv** keyword is available only when the crypto software image is installed. • (Optional) Enter **read** *readview* with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. • (Optional) Enter **write** *writeview* with a string (not to exceed 64 characters) that is the name of the view in which you can enter data and configure the contents of the agent. • (Optional) Enter **notify** *notifyview* with a string (not to exceed 64 characters) that is the name of the view in which you can specify a notify, inform request, or trap. • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
4	`router (config)# snmp-server user` *username groupname* [**remote** *host* [**udp-port** *port*]] {**v1** / **v2c** [**access** *access-list*]}	Configure a new user to an SNMP group. • The *username* is the name of the user on the host that connects to the agent. • The *groupname* is the name of the group with which the user is associated. • (Optional) Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity along with the

		optional UDP port number. The default UDP port number is 162. • Enter the SNMP version number (**v1** or **v2c**). • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
5	**router (config)# end**	Return to privileged EXEC mode.

Configuring SNMP Notifications

A trap manager is a management station that receives and processes notification types (traps). Traps are system alerts that the ML-Series card generates when certain events occur. By default, no trap manager is defined, and no traps are sent. To enable all traps, configure the **snmp-server enable traps command with no notification type keywords specified.**

Table 24-4 describes some of the more commonly used traps supported by the ML-Series card. You can enable any or all of these traps and configure a trap manager to receive them.

Table 24-4: ML-Series Card Notification Types

Notification Type Keyword	Description
bridge	Generates Spanning Tree Protocol (STP) bridge MIB traps.
config	Generates a trap for SNMP configuration changes.
config-copy	Generates a trap for SNMP copy configuration changes.
entity	Generates SNMP entity traps.
rsvp	Generates RSVP flow change traps.
rtr	Generates a trap for the SNMP Response Time Reporter (RTR).

You can send the **snmp-server host** global configuration command to a specific host to receive the notification types listed in Table 24-4.

Beginning in privileged EXEC mode, follow these steps to configure the ML-Series card to send traps or inform requests to a host:

Step	Command	Purpose
1	router#**configure terminal**	Enter global configuration mode.
2	router (config)# **snmp-server engineID remote** *ip-address engineid-string*	Specify the IP address and engine ID for the remote host.
3	router (config)# **snmp-server user** *username groupname* [**remote** *host*][**udp-port** *port*] {**v1** | **v2c**}[**access** *access-list*]	Configure an SNMP user to be associated with the remote host created in Step 2. • The *username* is the name of the user on the host that connects to the agent. • The *groupname* is the name of the group with which the user is associated. • (Optional) Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity along with the optional

		UDP port number. The default UDP port number is 162. • Enter the SNMP version number (**v1** or **v2c**). • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list. **Note:** You cannot configure a remote user for an address without first configuring the engine ID for the remote host. If you try to configure the user before configuring the remote engine ID, you receive an error message, and the command is not executed.
4	`router (config)# snmp-server host host-addr [traps | informs] [version 1 | 2c] community-string [udp-port port] [notification-type]`	Specify the recipient of an SNMP trap operation. • For *host-addr,* specify the name or Internet address of the host (the targeted recipient). • (Optional) Enter **traps** (the default) to send SNMP traps to the host. Enter **informs** to send SNMP inform requests to the host. • (Optional) Specify the SNMP **version** (**1 or 2c**). SNMPv1 does not support inform requests. • For *community-string*, enter the password-like community string sent with the notification operation. • (Optional) For **udp-port** *port*, enter the remote device UDP port. • (Optional) For *notification-type*, use the keywords listed in Table: ML-Series Card Notification Types. If no type is specified, all notifications are sent.
5	`router (config)#snmp-server enable traps notification-types`	Enable the ML-Series card to send traps or inform requests and specify the type of notifications to be sent. For a list of notification types, enter: **snmp-server enable traps ?** To enable multiple types of traps, you must enter a separate **snmp-server enable traps** command for each trap type.
6	`router (config)# snmp-server trap-source interface-id`	(Optional) Specify the source interface, which provides the IP address for the trap message. This command also sets the source IP address for inform requests.
7	`router (config)# snmp-server queue-length length`	(Optional) Establish how many trap messages each trap host can hold (message queue length.) The range is 1 to 1000; the default is 10.

8	`router (config)# snmp-server trap-timeout seconds`	(Optional) Define how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
9	**router (config)# end**	Return to privileged EXEC mode.

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the mechanism for the specified notification (for traps and inform requests). To enable a host to receive an inform request, you must configure an **snmp-server host informs** command for the host and globally enable inform requests by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not inform requests, to the host. To disable inform requests, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

Beginning in privileged EXEC mode, follow these steps to set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file:

Step	Command	Purpose
1	`router# configure terminal`	Enter global configuration mode.
2	`router (config)#snmp-server contact text`	Set the system contact string. For example: **snmp-server contact Dial System Operator at beeper 21555.**
3	`router (config)#snmp-server location text`	Set the system location string. For example: **snmp-server location Building 3/Room 222**
4	`router (config)# end`	Return to privileged EXEC mode.

Limiting TFTP Servers Used Through SNMP

Beginning in privileged EXEC mode, follow these steps to limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list:

Step	Command	Purpose
1	`router# configure terminal`	Enter global configuration mode.
2	`router (config)# snmp-server tftp-server-list access-list-number`	Limit TFTP servers used for configuration file copies through SNMP to the servers in the access list. For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
3	`router (config)# access-list access-list-number {deny | permit}`	Create a standard access list, repeating the command as many times as necessary.

	`source [source-wildcard]`	• For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the TFTP servers that can access the ML-Series card. • (Optional) For *source-wildcard*, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
4	`router (config)#` **`end`**	Return to privileged EXEC mode.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string "public".

This configuration does not cause the ML-Series card to send any traps.

```
ML_Series(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string "public". The ML-Series card also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2c. The community string "public" is sent with the traps.

```
ML_Series(config)# snmp-server community public

ML_Series(config)# snmp-server host 192.180.1.27 version 2c public

ML_Series(config)# snmp-server host 192.180.1.111 version 1 public

ML_Series(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the comaccess community string. No other SNMP managers have access to any objects. SNMP authentication failure traps are sent by SNMPv2c to the host cisco.com using the community string "public".

```
ML_Series(config)# snmp-server community comaccess ro 4

ML_Series(config)# snmp-server enable traps snmp authentication

ML_Series(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host cisco.com. The community string is restricted. The second line specifies the destination of these traps and overwrites any previous **snmp-server host** commands for the host cisco.com.

```
ML_Series(config)# snmp-server enable traps

ML_Series(config)# snmp-server host cisco.com restricted
```

This example shows how to enable the ML-Series card to send all traps to the host myhost.cisco.com using the community string "public".

```
ML_Series(config)# snmp-server enable traps

ML_Series(config)# snmp-server host myhost.cisco.com public
```

Displaying SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command.You can also use the other privileged EXEC commands in Table 24-5 to display SNMP information. For information about the fields in the output displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2.*

Table 24-5: Commands for Displaying SNMP Information

Feature	Default Setting
show snmp	Displays SNMP statistics.
show snmp group	Displays information about each SNMP group on the network.
show snmp pending	Displays information about pending SNMP requests.
show snmp sessions	Displays information about the current SNMP sessions.
show snmp user	Displays information about each SNMP user name in the SNMP users table.