Embedded Syslog Manager (ESM)

The Embedded Syslog Manager (ESM) feature provides a programmable framework that allows you to filter, escalate, correlate, route, and customize system logging messages prior to delivery by the Cisco IOS system message logger.

Feature History for the Embedded Syslog Manager Feature

Release	Modification
12.3(2)T	This feature was introduced.
12.3(2)XE	This feature was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About the Embedded Syslog Manager

To configure the Embedded Syslog Manager, you should understand the following concepts:

- Cisco IOS System Message Logging, page 2
- System Logging Message Formatting, page 2
- The Embedded Syslog Manager, page 3
- Syslog Filter Modules, page 3

Cisco IOS System Message Logging

The Cisco IOS system message logging (syslog) process allows the system to report and save important error and notification messages, either locally or to a remote logging server. These syslog messages include messages in a standardized format (called system logging messages, system error messages, or simply system messages) and output from **debug** commands. These messages are generated during network operation to assist users and Cisco TAC engineers with identifying the type and severity of a problem, or to aid users in monitoring router activity. System logging messages can be sent to console connections, monitor (TTY) connections, the system buffer, or to remote hosts.

With the introduction of the Embedded Syslog Manager, system messages can be logged independently as standard messages, XML-formatted messages, or ESM filtered messages. These outputs can be sent to any of the traditional syslog targets. For example, you could enable standard logging to the console connection, XML-formatted message logging to the buffer, and ESM filtered message logging to the monitor. Similarly, each type of output could be sent to different remote hosts. A benefit of separate logging processes is that if, for example, there is some problem with the ESM filter modules, standard logging will not be affected.

System Logging Message Formatting

System logging messages take the following format:

%<facility>-<severity>-<mnemonic>: <message-text>

For example:

```
%LINK-5-CHANGED: Interface Serial3/3, changed state to administratively down
```

Usually, these messages are proceeded by additional text, such as the timestamp and error sequence number:

<sequence-number>: <timestamp>:%<facility>-<severity>-<mnemonic>: <message-text>

For example:

```
000013: Mar 18 14:52:10.039:%LINK-5-CHANGED: Interface Serial3/3, changed state
to administratively down
```

Note The timestamp format used in system logging messages is determined by the **service timestamps** global configuration mode command. The **service sequence-numbers** global configuration command enables or disables the leading sequence number. An asterisk (*) before the time indicates that the time may be incorrect because the system clock has not synchronized to a reliable time source.

The Embedded Syslog Manager

The Embedded Syslog Manager (ESM) is a feature integrated in Cisco IOS software that allows complete control over system message logging at the source. ESM provides a programmatic interface to allow you to write custom filters that meet your specific needs in dealing with system logging. Benefits of this feature include:

- Customization—Fully customizable processing of system logging messages, with support for multiple, interfacing syslog collectors.
- Severity escalation for key messages—The ability to configure your own severity levels for syslog messages instead of using the system-defined severity levels.
- Specific message targeting—The ability to route specific messages or message types, based on type of facility or type of severity, to different syslog collectors.
- SMTP-base email alerts—Capability for notifications using TCP to external servers, such as TCP-based syslog collectors or Simple Mail Transfer Protocol (SMTP) servers.
- Message Limiting—The ability to limit and manage syslog “message storms” by correlating device-level events.

The ESM is not a replacement for the current UDP-based syslog mechanism; instead, it is an optional subsystem that can operate in parallel with the current system logging process. For example, you can continue to have the original syslog message stream collected by server A, while the filtered, correlated, or otherwise customized ESM logging stream is sent to server B. All of the current targets for syslog messages (console, monitor, buffer, and syslog host list) can be configured to receive either the original syslog stream or the ESM stream. The ESM stream can be further divided into user-defined streams and routed to collectors accordingly.

Syslog Filter Modules

To process system logging messages, the ESM uses syslog filter modules. Syslog filter modules are merely scripts written in the Tcl script language stored in local system memory or on a remote file server. The ESM is customizable because you can write and reference your own scripts.

Syslog filter modules can be written and stored as plain-text files or as precompiled files. Tcl script pre-compiling can be done with tools such as TclPro. Precompiled scripts allow a measure of security and managed consistency because they cannot be edited.

Note As Tcl script modules contain executable commands, you should manage the security of these files in the same way you manage configuration files.

Restrictions for Embedded Syslog Manager

ESM depends upon the Tcl 8.3.4 Cisco IOS subsystem, as ESM filters are written in Tcl. ESM is only available in images that support Tcl version 8.3.4 or later. Support for Tcl 8.3.4 is introduced in Cisco IOS Release 12.3(2)T.

ESM filters are written in Tcl. This document assumes the reader is familiar with Tcl programming.

ESM filtering cannot be applied to SNMP “history” logging. In other words, ESM filtering will not be applied to messages logged using the **logging history** and **snmp-server enable traps syslog** commands.

How to Use the Embedded Syslog Manager

To use the Embedded Syslog Manager, perform the following tasks:

- Writing ESM Syslog Filter Modules, page 4
- Configuring the Embedded Syslog Manager, page 10

Writing ESM Syslog Filter Modules

Before referencing syslog filter modules in the ESM configuration, you must write or obtain the modules you wish to apply to system logging messages. Syslog filter modules can be stored in local system memory, or on a remote file server. To write syslog filter modules, you should understand the following concepts:

- The ESM Filter Process, page 4
- Syslog Filter Module Input, page 4
- Normal ESM Filter Processing, page 7
- Background ESM Filter Processing, page 8

The ESM Filter Process

When ESM is enabled, all system logging messages are processed through the referenced syslog filter modules. Syslog filter modules are processed in their order in the filter chain. The position of a syslog filter module in the filter chain is determined by the position tag applied in the **logging filter** global configuration mode command. If a position is not specified, the modules are processed in the order in which they were added to the configuration.

The output of each filter module is used as the input for the next filter module in the chain. In other words, the Tcl global variable containing the original syslog message (::orig_msg) is set to the return value of each filter before calling the next filter in the chain. Thus, if a filter returns NULL, no message will be sent out to the ESM stream. Once all filters have processed the message, the message is enqueued for distribution by the logger.

The console, buffer, monitor, and syslog hosts can be configured to receive a particular message stream (normal, XML, or ESM). The syslog hosts can be further restricted to receive user-defined numbered streams. Each target examines each message and accepts or rejects the message based on its stream tag. ESM filters can change the destination stream by altering the messages' stream tag by changing the Tcl global variable "::stream".

Syslog Filter Module Input

When ESM is enabled, system logging messages are sent to the logging process. Each of the data elements in the system logging message, as well as the formatted syslog message as a whole, are recorded as Tcl global variables. The data elements for the syslog message are as follows:

<sequence-number>: <timestamp>:%<facility>-<severity>-<mnemonic>: <message-text>

The message-text will often contain message-arguments.

When messages are received on a syslog host a "syslog-count" number is also added:

<syslog-count>: <sequence-number>: <timestamp>:%<facility>-<severity>-<mnemonic>: <message-text>

For example:

```
24:000024:02:18:37:%SYS-5-CONFIG_I:Configured from console by console
```

Table 1 lists the Tcl script input variables used in syslog filter modules. The syslog message data that the filter must operate on are passed as Tcl global namespace variables. Therefore, variables should be prefixed by a double-colon within the script module.

Table 1 Valid Variables for Syslog Filter Modules

Variable Name	Definition
::orig_msg	Full original system logging message as formatted by the system. • If the filter module is just making decisions on whether to send a message or not, return either NULL or the value of this variable ($::orig_msg)
::hostname	The router's hostname. • The hostname can be added to the beginning of syslog messages sent to remote hosts using the **logging origin-id hostname** global configuration mode command.
::buginfseq	The error message sequence-number. • The **service sequence-numbers** global configuration command enables or disables the leading sequence number.
::timestamp	The timestamp on the system logging message. • The timestamp format used in system logging messages is determined by the **service timestamps** global configuration mode command.
::facility	The name of the system facility that generated the message. • The FACILITY is a code consisting of two or more uppercase letters that indicate the facility to which the message refers. A facility can be a hardware device, a protocol, or a module of the system software. Common examples include SYS, LINK, LINEPROTO, and so on.
::severity	The severity value. • The SEVERITY is a single-digit code from 0 to 7 that reflects the severity of the condition. The lower the number, the more serious the message. • The syslog filter module should change this variable if the severity is to be escalated.
::mnemonic	The message mnemonic. • The MNEMONIC is a code (usually an abbreviated description) that uniquely identifies the type of error or event. Common examples include CONFIG_I, UPDOWN, and so on.
::format_string	The message-text string. • The format string is used to create the original message. The message text will often contain arguments; for example, in the message "Configured from %s by %s", %s indicates the message arguments. • The message-text string is the message form that can be passed to the Tcl **format** command.

Table 1 Valid Variables for Syslog Filter Modules

Variable Name	Definition
::msg_args	The message-text arguments. • The msg_args variable is the list containing the arguments for the format_string. • For example, in the system logging message "2w0d: %SYS-5-CONFIG_I: Configured from console by console." the format_string is "Configured from %s by %s." and the msg_args are "console, console".
::process	The process name and interrupt level string. • Some system messages describe internal errors and contain traceback information. The following sample output shows the format for process and interrupt level (ipl) information: `-Process= "Net Background", ipl= 2, pid= 82`
::pid	The process ID (PID). • Some system messages include the process ID of the triggering process. The following sample output shows the format for process ID (pid) information: `-Process= "Net Background", ipl= 2, pid= 12345`

::traceback	The traceback string.
	• Some system messages describe internal errors and contain traceback information. This information, when included, will typically appear at the end of an error message. The following sample output shows the format for traceback information:
	`Apr 23 07:14:02: %ATMPA-3-CMDFAIL: ATM2/1/0 Command Failed`
	`at ../src-rsp/rsp_vip _atmdx.c - line 113, arg 32784`
	`-Process= "Net Background", ipl= 2, pid= 82`
	`-Traceback= 602D12AC 602CED14 60050B6C 602CFF74`

::syslog_facility	The syslog facility number used in the PRI portion of the syslog message sent to external syslog collectors (syslog hosts). • The syslog facility is given as a number, from 0 to 184. • The default is 184 (local7), but the value can be changed with the **logging facility** global configuration command.
::clear	Contains the string "- event cleared" or "NULL".
::version	The Cisco IOS software version, in the format "SYS_MAJORVERSION. SYS_MINORVERSION".

Table 1 Valid Variables for Syslog Filter Modules

Variable Name	Definition
::module_position	The position of this syslog filter module in the filter chain. The filter chain starts at one (1). • The value of this argument is determined by the order in which the scripts are referenced by the **logging filter** global configuration mode command.
::stream	The ESM message stream number. • The stream number will always be set to 2 (filtered stream) prior to the first filter being executed. • Syslog filter modules can change this value to a user-defined stream number in order to route the message to particular syslog collectors. • Stream numbers are allocated as follows: – Stream 0: Default (standard) syslog stream – Stream 1: XML tagged syslog stream – Stream 2: Default filtered syslog stream – Streams 3-9: Reserved – Streams 10-65536: User defined

Normal ESM Filter Processing

Each time a system logging message is generated, the syslog filter modules are called in a series. This series is determined by the ::module_position variable, which in turn is typically the order in which the modules are referenced in the system configuration (the order in which they are configured).

The output of one filter module becomes the input to the next. Because the input to the filters are the Tcl global namespace variables (as listed in Table 1), each filter can change any or all of these variables depending upon the purpose of the filter.

The only Tcl global variables that are automatically updated by the ESM framework between subsequent filter executions are the ::orig_msg and ::cli_args variables. The framework automatically sets the value of ::orig_msg to the return value of the filter module. Thus a filter that is designed to alter or filter the original message must not manually set the value for the ::orig_msg variable; the filter only needs to return the desired value.   For example, the following one-line ESM filter

```
return “This is my new syslog message.”
```

would ignore any message passed to it, and always change the output to the constant string “This is my new syslog message.” If the module was the last filter in the chain, all ESM targets would receive this string as the final syslog message.

The one-line ESM filter

```
return “”
```

would block all syslog messages to the ESM stream. For example, the line

```
return $::orig_msg
```

would do nothing but pass the message along to the next filter in the chain. Thus, an ESM filter designed to suppress unwanted messages would look something like this:

```
if { [my_procedure_to_check_this_message] == 1 } {
    return $::orig_msg
} else {
    return “”
}
```

Depending upon their design, some filters may not use the ::orig_msg variable at all, but rather reconstruct a syslog message from its data elements (using ::format_string, ::msg_args, ::timestamp, and so on). For example, an XML tagging filter will tag the individual data elements, and disregard the original formatted message. It is important for such modules to check the ::orig_msg variable at the beginning of the Tcl script, so that if previous filter indicated that the message should not be sent out (::orig_msg is NULL), it would not bother to process the message, but simply return NULL also.

Cisco IOS commands can also be added to syslog filter modules using the **exec** and **config** Tcl commands. For example, if you wanted to add the source IP address to the syslog messages, and syslog messages were configured to be sent from the Ethernet 2/0 interface (using the **logging source-interface** command) you could issue the **show interface Ethernet 2/0** command during the module initialization by using the **exec** Tcl command within the script:

```
set source_ip_string [exec show ip int E2/0 | inc Internet]

puts $source_ip_string

"  Internet address is 10.4.2.63/24"
```

The script should then pass the output of that command to the syslog message. For further information on scripting within Cisco IOS software, see the “Cisco IOS Scripting with Tcl” feature guide document on Cisco.com.

Background ESM Filter Processing

In Tcl it is possible to queue commands for processing in the future by using the **after** Tcl command. The most common use of this command is to correlate (gather and summarize) events over a fixed interval of time, called the “correlation window”. Once the window of interest expires, the filter will need to “wake up”, and calculate or summarize the events that occurred during the window, and often send out a new syslog message to report the events. This background process is handled by the ESM Event Loop process, which allows the Tcl interpreter to execute queued commands after a certain amount of time has passed.

If your syslog filter module needs to take advantage of correlation windows, it must use the **after** Tcl command to call a summary procedure once the correlation window expires (see examples in Appendix A). Since there is no normal filter chain processing when background processes are run, in order to produce output these filters must make use of one of two ESM Tcl extensions: **errmsg** or **esm_errmsg**.

During background processing, the commands that have been enqueued by the **after** command are not run in the context of the filter chain (as in normal processing), but rather are autonomous procedures that are executed in series by the Tcl interpreter.   Thus, these background procedures should not operate on the normal Tcl global namespace variables (except for setting the globals for the next filter when using **esm_errmsg**), but should operate on variables stored in their own namespace. If these variables are declared outside of a procedure definition, they will be persistent from call to call.

The purpose of the **errmsg** Tcl command is to create a new message and send it out for distribution, bypassing any other syslog filter modules. The syntax of the **errmsg** command is:

```
errmsg <severity> <stream> <message_string>
```

The purpose of the **esm_errmsg** Tcl command is to create a new message, process it with any syslog filter modules below it in the filter chain, and then send it out for distribution. The syntax of the **esm_errmsg** command is:

```
esm_errmsg <module_position>
```

The key difference between the errmsg() Tcl function and the esm_errmsg() Tcl function is that **errmsg** ignores the filters and directly queues a message for distribution, while **esm_errmsg** will send a syslog message down the chain of filters.

In the following example, a new syslog message is created and sent out tagged as Alert severity 1 to the configured ESM logging targets (stream 2). One can assume the purpose of this filter would be to suppress the individual SYS-5-CONFIG messages over a thirty minute correlation window, and send out a summary message at the end of the window.

```
errmsg 1 2 "*Jan 24 09:34:02.539: %SYS-1-CONFIG_I: There have been 12
configuration changes to the router between Jan 24 09:04:02.539 and Jan 24
09:34:01.324"
```

In order to use **esm_errmsg**, since the remaining filters below this one will be called, this background process must populate the needed TCL global namespace variables prior to calling **esm_errmsg**. Passing the ::module_position tells the ESM framework which filter to start with. Thus, filters using the **esm_errmsg** command should store their ::module_position (passed in the global during normal processing) in their own namespace variable for use in background processing. Here is an example:

```
proc ::my_filter_namespace::my_summary_procedure{}
{
 set ::orig_msg "*Jan 24 09:34:02.539: %SYS-1-CONFIG_I: There have been 12
configuration changes to the router between Jan 24 09:04:02.539 and Jan 24
09:34:01.324"
 set ::timestamp "*Jan 24 09:34:02.539"
 set ::severity 1
 set ::stream 2
 set ::traceback ""
 set ::pid ""
 set ::process ""
 set ::format_string "There have been %d configuration changes to the router
between %s and %s"
 set ::msg_args {12 "Jan 24 09:04:01.539" "Jan 24 09:34:01.324"}
 esm_errmsg $::my_filter_namespace::my_module_position
}
```

The benefit of setting all the global variables for the **esm_errmsg** command is that your filters will be modular, and it will not matter what order they are used in the ESM framework. For example, if you wish all of the messages destined for the ESM targets to suffixed with the message originator's hostname, you could write a one-line "hostname" filter and place it at the bottom of the filter chain:

```
return "$::orig_msg -- $::hostname"
```

In this example, if any of your filters generate new messages during background processing and they use **esm_errmsg** instead of **errmsg**, these messages will be clearly suffixed with the hostname.

What to Do Next

After creating your syslog filter module, you should store the file in a location accessible to the router. You can copy the file to local system memory, or store it on a network file server.

Configuring the Embedded Syslog Manager

To configure the ESM, specify one or more filters to be applied to generated syslog messages, and specify the syslog message target.

Prerequisites

One or more syslog filter modules must be available to the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **logging filter** *filter-url*
4. Repeat Step 3 for each syslog filter module that should be applied to system logging output.
5. **logging** [**console** | **buffered** | **monitor**] **filtered** [*level*]

   or

   logging host {*ip-address* | *host-name*} **filtered** [**stream** *stream-id*]
6. Repeat Step 5 for each desired system logging destination.
7. **logging source-interface**
8. **logging origin-id**
9. **end**
10. **show logging**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**`logging filter`** *`filter-url`* `[`*`position`*`] [`**`args`** *`filter-arguments`*`]`
	Example:
	`Router(config)# logging filter slot0:/escalate.tcl 1 args CONFIG_I 1`
		Specifies one or more syslog filter modules to be applied to generated system logging messages.
		- Repeat this command for each syslog filter module that should be used.
		- The *filter-url* argument is the Cisco IOS File System location of the syslog filter module (script). The location can be in local memory, or a remote server using **tftp:**, **ftp:**, or **rcp:**.
		- The optional *position* argument specifies the order in which the syslog filter modules should be executed. If this argument is omitted, the specified module will be positioned as the last module in the chain.
		- Filters can be re-ordered on the fly by re-entering the **logging filter** command and specifying a different position.
		- The optional **args** *filter-arguments* syntax can be added to pass arguments to the specified filter. Multiple arguments can be specified. The number and type of arguments should be defined in the syslog filter module. For example, if the syslog filter module is designed to accept a specific email address as an argument, you could pass the email address using the **args user@host.com** syntax.Multiple arguments are typically delimited by spaces.
		- To remove a module from the list of modules to be executed, use the **`no`** form of this command.

Step 4	Repeat Step 3 for each syslog filter module that should be applied to system logging output.	—

	Command or Action	Purpose

Step 5	**logging** [**console** | **buffered** | **monitor**] **filtered** [*level*]
	or
	logging host {*ip-address* | *host-name*} **filtered** [**stream** *stream-id*]
	Example:
	`Router(config)# logging console filtered informational`
	or
	`Router(config)# logging host 209.165.200.225 filtered stream 20`
		Specifies the target for ESM filtered syslog output.
		• ESM filtered syslog messages can be sent to the console, a monitor (TTY and Telnet connections), the system buffer, or to remote hosts.
		• The optional *level* argument limits the sending of messages to those at or numerically lower than the specified value. For example, if level **1** is specified, only messages at level 1 (alerts) or level 0 (emergencies) will be sent to the specified target. The level can be specified as a keyword or number.
		• When logging to the console, monitor connection, or system buffer, the severity threshold specified by the *level* argument takes precedence over the ESM filtering. In other words, even if the ESM filters return a message to be delivered to ESM targets, if the severity doesn't meet the configured threshold (is numerically higher than the level value), it will not be delivered.
		• When logging to remote hosts, the stream tag allows you to specify a destination based on the type of message. The **stream** *stream-id* syntax allows you to configure the ESM to send only messages that have a specified stream value to a certain host.
		• The stream value is applied to messages by the configured syslog filter modules. For example, all Severity 5 messages could have a stream tag of "20" applied. You can then specify that all messages with a stream tag of "20" be sent to the host at 209.165.200.225 using the command:
		logging host 209.165.200.225 filtered stream 20

Step 6	Repeat Step 5 for each desired system logging destination.	• By issuing the logging host command multiple times, you can specify different targets for different system logging streams. • Similarly, you can configure messages at different severity levels to be sent to the console, monitor connection, or system buffer. For example, you may want to display only very important messages to the screen (using a monitor or console connection) at your network operations center (NOC).
Step 7	**logging source-interface** *type number*	(Optional) Specifies the source interface for syslog messages sent to remote syslog hosts. • Normally, a syslog messages sent to remote hosts will use whatever interface is available at the time of the message generation. This command forces the router to send syslog messages to remote hosts only from the specified interface.

	Command or Action	Purpose

Step 8	**logging origin-id** {**hostname** | **ip** | **string** *user-defined-id*}
	Example:
	`Router(config)# logging origin-id string “Domain 2, Router 5”`
		(Optional) Allows you to add an origin identifier to syslog messages sent to remote hosts.
		• The origin identifier is added to the beginning of all syslog messages sent to remote hosts. The identifier can be the hostname, the IP address, or any text that you specify.
		• The origin identifier is useful for identifying the source of system logging messages in cases where you send syslog output from multiple devices to a single syslog host.

Step 9	**end**
	Example:
	`Router(config)# end`
		Ends your current configuration session and returns the CLI to privileged EXEC mode.

Step 10	**show logging** **Example:** `Router# show logging`	(Optional) Displays the status of system logging, including the status of ESM filtered logging. • If filtered logging to the buffer is enabled, this command also shows the data stored in the buffer. • The order in which syslog filter modules are listed in the output of this command is the order in which the filter modules are executed.

Configuration Examples for the Embedded Syslog Manager

In the following example, ESM filter logging is enabled for the console connection, standard logging is enabled for the monitor connection and for the buffer, and XML-formatted logging is enabled for the host at 209.165.200.225:

```
Router(config)# logging filter tftp://209.165.200.225/ESM/escalate.tcl
Router(config)# logging filter slot0:/email.tcl user@example.com
Router(config)# logging filter slot0:/email_guts.tcl
Router(config)# logging console filtered
Router(config)# logging monitor 4
Router(config)# logging buffered debugging
Router(config)# logging host 209.165.200.225 xml
Router(config)# end
Router# show logging
Syslog logging: enabled (0 messages dropped, 8 messages rate-limited,
               0 flushes, 0 overruns, xml disabled, filtering enabled)
    Console logging: level debugging, 21 messages logged, xml disabled,
                     filtering enabled
    Monitor logging: level warnings , 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 30 messages logged, xml disabled,
                     filtering disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
```

```
Filter modules:
    tftp://209.165.200.225/ESM/escalate.tcl
    slot0:/email.tcl user@example.com

    Trap logging: level informational, 0 message lines logged
        Logging to 209.165.200.225, 0 message lines logged, xml enabled,
              filtering disabled

Log Buffer (8192 bytes):

*Jan 24 09:34:28.431: %SYS-5-CONFIG_I: Configured from console by console
*Jan 24 09:34:51.555: %SYS-5-CONFIG_I: Configured from console by console
*Jan 24 09:49:44.295: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Additional References

For additional information related to Embedded Syslog Manager, refer to the following references:

Related Documents

Related Topic	Document Title
System Message Logging	"Troubleshooting and Fault Management" chapter of the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3.
XML Formatted System Message Logging	"XML Interface to Syslog Messages," Cisco IOS Release 12.2(15)T feature module
System Logging Messages	*Cisco IOS Software System Messages*, Release 12.3
Tcl 8.3.4 Support in Cisco IOS Software	"Cisco IOS Scripting with Tcl," Cisco IOS Release 12.3(2)T feature module

Standards

Standards	Title
None	—

MIBs

MIBs[1]	MIBs Link
- None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

[1] Not all supported MIBs are listed.

RFCs

RFCs[1]	Title
RFC-3164	The BSD Syslog Protocol - This RFC is informational only. The Cisco implementation of syslog does not claim full compliance with the protocol guidelines mentioned in this RFC.

[1] Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new and modified commands only.

- **logging buffered filtered**
- **logging console filtered**
- **logging filter**
- **logging host**
- **logging monitor filtered**
- **logging origin-id**
- **show logging**

logging buffered filtered

To enable Embedded Syslog Manager (ESM) filtered system message logging to the standard syslog buffer, use the **logging buffered filtered** command in global configuration mode. To disable all logging to the buffer and return the size of the buffer to the default, use the **no** form of this command.

logging buffered filtered [*severity-level*]

no logging buffered filtered

Syntax Description

severity-level	(Optional) Limits messages sent to the buffer to those messages at or numerically lower than the specified value. For example, if level **1** is specified, only messages at level 1 (alerts) or level 0 (emergencies) will be sent to the specified target. Severity levels are specified as a number or a keyword: {**0** | **emergencies**}—System is unusable {**1** | **alerts**}—Immediate action needed {**2** | **critical**}—Critical conditions {**3** | **errors**}—Error conditions {**4** | **warnings**}—Warning conditions {**5** | **notifications**}—Normal but significant conditions {**6** | **informational**}—Informational messages {**7** | **debugging**}—Debugging messages

Defaults

Logging to the buffer is enabled.

ESM filtering of system logging messages sent to the buffer is disabled.

The default severity level varies by platform but is generally level 7 (“debugging”), meaning that messages at all severity levels (0 through 7) are logged.

Command Modes

Global configuration

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

If standard logging has been disabled on your system (using the **no logging on** command), standard logging must be reenabled using the **logging on** command before using the **logging buffered filtered** command.

Standard logging is enabled by default, but filtering by the ESM is disabled by default.

ESM uses syslog filter modules, which are Tcl script files stored locally or on a remote device. The syslog filter modules must be configured using the **logging filter** command before filtered output can be sent to the buffer.

When ESM filtering is enabled, all messages sent to the buffer have the configured syslog filter modules applied. To return to standard logging to the buffer, use the plain form of the **logging buffered** command (without the **filtered** keyword). To disabled all logging to the buffer, use the **no logging buffered** command, with or without the **filtered** keyword.

The buffer is circular, so newer messages overwrite older messages as the buffer is filled. To change the size of the buffer, use the **logging buffered** *buffer-size* command, then issue the **logging buffered filtered** command to start (or restart) filtered logging.

To display the messages that are logged in the buffer, use the **show logging** command in EXEC mode. The first message displayed is the oldest message in the buffer.

Examples

In the following example, the user enables ESM filtered logging to the buffer:

```
Router(config)# logging filter tftp://209.165.200.225/ESM/escalate.tcl
Router(config)# logging filter slot0:/email.tcl user@example.com
Router(config)# logging buffer filtered
```

Related Commands

Command	Description
clear logging	Clears all messages from the system message logging (syslog) buffer.
logging buffered	Enables standard system message logging (syslog) to a local buffer and sets the severity level and buffer size for the logging buffer.
logging filter	Specifies the name and location of a syslog filter module to be applied to generated system logging messages.
logging on	Globally controls (enables or disables) system message logging.
show logging	Displays the state of system message logging, followed by the contents of the logging buffer.

logging console filtered

To enable Embedded Syslog Monitor (ESM) filtered system message logging to the console connections, use the **logging console filtered** command in global configuration mode. To disable all logging to the console connections, use the **no** form of this command.

logging console filtered [*severity-level*]

no logging console [**filtered**] [s*everity-level*]

Syntax Description

severity-level	(Optional) The number or name of the desired severity level at which messages should be logged. Messages at or numerically lower than the specified level are logged. Severity levels are as follows (enter the number or the keyword): {**0** | **emergencies**}—System is unusable {**1** | **alerts**}—Immediate action needed {**2** | **critical**}—Critical conditions {**3** | **errors**}—Error conditions {**4** | **warnings**}—Warning conditions {**5** | **notifications**}—Normal but significant conditions {**6** | **informational**}—Informational messages {**7** | **debugging**}—Debugging messages

Defaults

Logging to the console is enabled.

ESM filtering of system logging messages sent to the console is disabled.

The default severity level varies by platform, but is generally level 7 (messages at levels 0 through 7 are logged).

Command Modes

Global configuration

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

If standard logging has been disabled on your system (using the **no logging on** command), standard logging must be reenabled using the **logging on** command before using the **logging console filtered** command.

Standard logging is enabled by default, but filtering by the ESM is disabled by default.

ESM uses syslog filter modules, which are Tcl script files stored locally or on a remote device. The syslog filter modules must be configured using the **logging filter** command before system logging messages can be filtered.

When ESM filtering is enabled, all messages sent to the console have the configured syslog filter modules applied. To disable filtered logging to the console and return to standard logging, use the standard **logging console** command (without the **filtered** keyword). To disable all logging to the console, use the **no logging console** command, with or without the **filtered** keyword.

Examples

In the following example, the user enables ESM filtered logging to the console for severity levels 0 through 3:

```
Router(config)# logging filter tftp://209.165.200.225/ESM/escalate.tcl
Router(config)# logging filter slot0:/email.tcl user@example.com
Router(config)# logging console filtered 3
```

Related Commands

Command	Description
logging console	Enables standard system message logging (syslog) to all console (CTY) connections and sets the severity level.
logging filter	Specifies the name and location of a syslog filter module to be applied to generated system logging messages.
logging on	Globally controls (enables or disables) system message logging.
show logging	Displays the state of system message logging, followed by the contents of the logging buffer.

logging filter

To specify a syslog filter module to be used by the Embedded Syslog Manager (ESM), use the **logging filter** command in global configuration mode. To remove a module from the filter chain, use the **no** form of this command.

logging filter *filter-url* [*position*] [**args** *filter-arguments*]

no logging filter *filter-url* [*position*]

Syntax Description

filter-url	Specifies the location of the syslog filter module (script file), using the standard Cisco IOS File System URL syntax. • The location can be a local memory location, such as **flash:** or **slot0:**, or a remote file server system, such as **tftp:**, **ftp:**, or **rcp:**. • The *filter-url* should include the name of the syslog filter module; for example, “email.tcl” or “email.txt”.
position	(Optional) An integer that specifies the order in which the syslog filter modules should be executed. The valid value for this argument is N + 1, where N is the current number of configured filters. • If this argument is omitted, the specified module will be positioned as the last module in the chain (the Nth+1 position).
args *filter-arguments*	(Optional) Any arguments you wish to pass to the ESM file chain can be added using this syntax. The ESM filter modules will determine what arguments you should use.

Defaults

No ESM filters are applied to system logging messages.

Command Modes

Global configuration

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

Use this command to enable the Embedded Syslog Manager by specifying the filter that should be applied to logging messages generated by the system. Repeat this command for each syslog filter module that should be used.

Syslog filter modules are Tcl script files. These files can be stored as plain text files (.txt) or as precompiled Tcl scripts (.tcl). When positioning (ordering) the modules, keep in mind that the output of each filter module is used as input for the next filter module in the chain.

By default, syslog filter modules are executed in the order in which they appear in the system configuration file. The *position* argument can be used to order the filter modules manually. Filter modules can also be reordered at any time by reentering the **logging filter** command and specifying a different position for a given filter module.

The optional **args** *filter-arguments* syntax can be added to pass arguments to the specified filter. Multiple arguments can be specified. The number and type of arguments should be defined in the syslog filter module. For example, if the syslog filter module is designed to accept a specific email address as an argument, you could pass the email address using the **args user@host.com** syntax. Multiple arguments are typically delimited by spaces.

To remove a module from the list of modules to be executed, use the **no** form of this command. Modules not referenced in the configuration will not be executed, regardless of their "position" number.

Examples

In the following example, the user enables ESM filtered logging to the console for severity levels 0 through 3:

```
Router(config)# logging filter tftp://209.165.200.225/ESM/escalate.tcl
Router(config)# logging filter slot0:/email.tcl user@example.com
Router(config)# logging filter slot0:/email_guts.tcl
Router(config)# logging console filtered 3
```

Related Commands

Command	Description
logging buffer filtered	Enables ESM filtered system message logging to the system logging buffer.
logging console filtered	Enables ESM filtered system message logging to all console connections.
logging host	Enables system message logging to a remote host (syslog collector).
logging monitor filtered	Enables ESM filtered system message logging to all monitor (TTY) connections.
show logging	Displays the status of system message logging, followed by the contents of the logging buffer.

logging host

To log system messages and debug output to a remote host, use the **logging host** command in global configuration mode. To remove a specified logging host from the configuration, use the **no** form of this command.

logging host {*ip-address* | *hostname*} [**xml** | **filtered** [**stream** *stream-id*]]

no logging host {*ip-address* | *hostname*} [**xml** | **filtered** [**stream** *stream-id*]]

Syntax Description

ip-address	IP address of the host that will receive the system logging messages.
hostname	Name of the host that will receive the system logging messages.
xml	(Optional) Specifies that the logging output should be tagged using the Cisco defined XML tags.
filtered	(Optional) Specifies that logging messages sent to this host should first be filtered by the ESM syslog filter modules specified in the **logging filter** commands.
stream *stream-id*	(Optional) Specifies that only ESM filtered messages with the stream identification number specified in the *stream-id* argument should be sent to this host. (The *stream-id* number is applied to messages by syslog filter modules.)

Defaults

System logging messages are not sent to any remote host.

If this command is entered without the **xml** or **filtered** keywords, messages are sent in the standard format.

Command Modes

Global configuration

Command History

Release	Modification
10.0	The **logging** command was introduced.
12.0(14)S, 12.0(14)ST, 12.2(15)T	The **logging host** command replaced the **logging** command.
12.2(15)T	The **xml** keyword was added.
12.3(2)T	The **filtered** [**stream** *stream-id*] syntax was added as part of the Embedded Syslog Manager feature.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

Standard system message logging (syslog) is enabled by default. If logging has been disabled on your system (using the **no logging on** command), logging must be reenabled using the **logging on** command before using the **logging host** command.

The **logging host** command identifies a remote host (usually a device serving as a syslog server) to receive logging messages. By issuing this command more than once, you can build a list of hosts that receive logging messages.

To specify the severity level for logging to all hosts, use the **logging trap** command.

If XML-formatted syslog is enabled using the **logging host** {*ip-address* | *hostname*} **xml** command, messages will be sent to the specified host with the system defined XML tags. These tags are predefined and are not user-configurable. XML-formatting will not be applied to debugging output.

If you are using the Embedded Syslog Manager (ESM) feature, you can enable ESM filtered syslog messages to be sent to one or more hosts using the **logging host** {*ip-address* | *hostname*} **filtered** command. To use the ESM feature, you must first specify the syslog filter modules that should be applied to the messages using the **logging filter** command. See the description of the **logging filter** command for more information on the ESM feature.

To configure standard logging to a specific host after configuring XML-formatted or ESM filtered logging to that host, use the standard form of this command (**logging host** {*ip-address* | *hostname*}) without the **xml** or **filtered** keywords. In other words, a standard **logging host** command will replace an XML or ESM filtered **logging host** command, and vice versa, if the same host is specified.

Note Any **no logging host** command (with or without the optional keywords) will disable all logging to the specified host.

You can configure the system to send standard messages to one or more hosts, XML-formatted messages to one or more hosts, and ESM filtered messages to one or more hosts by repeating this command as many times as desired with the appropriate syntax. (See the “Examples” section.)

Examples

In the following example, messages at severity levels 0 (emergencies) through 5 (notifications) are logged to a host at 209.165.202.169:

```
Router(config)# logging host 209.165.202.169
Router(config)# logging trap 5
```

In the following example, standard system logging messages are sent to the host at 209.165.200.225, XML-formatted system logging messages are sent to the host at 209.165.200.226, ESM filtered logging messages with the stream 10 value are sent to the host at 209.165.200.227, and ESM filtered logging messages with the stream 20 value are sent to host at 209.165.202.129:

```
Router(config)# logging host 209.165.200.225
Router(config)# logging host 209.165.200.226 xml
Router(config)# logging host 209.165.200.227 filtered stream 10
Router(config)# logging host 209.165.202.129 filtered stream 20
```

Related Commands

Command	Description
logging on	Globally controls (enables or disables) system message logging.
logging trap	Limits messages sent to the syslog servers based on severity level.
show logging	Displays the state of system message logging, followed by the contents of the standard syslog buffer.
show logging xml	Displays the state of XML-formatted system message logging, followed by the contents of the XML syslog buffer.

logging monitor filtered

To enable Embedded Syslog Manager (ESM) filtered system message logging to monitor connections, use the **logging monitor filtered** command in global configuration mode. To disable all logging to the monitor connections, use the **no** form of this command.

logging monitor filtered [*severity-level*]

no logging monitor filtered

Syntax Description

severity-level	(Optional) The number or name of the desired severity level at which messages should be logged. Messages at or numerically lower than the specified level are logged. Severity levels are as follows (enter the number or the keyword): {**0** | **emergencies**}—System is unusable {**1** | **alerts**}—Immediate action needed {**2** | **critical**}—Critical conditions {**3** | **errors**}—Error conditions {**4** | **warnings**}—Warning conditions {**5** | **notifications**}—Normal but significant conditions {**6** | **informational**}—Informational messages {**7** | **debugging**}—Debugging messages

Defaults

Logging to monitor connections is enabled.

ESM filtering of system logging messages sent to the monitor connections is disabled.

The default severity level varies by platform, but is generally level 7 (messages at levels 0 through 7 are logged).

Command Modes

Global configuration

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

The **monitor** keyword specifies the TTY (TeleTYpe) line connections at all line ports. TTY lines (also called ports) communicate with peripheral devices such as terminals, modems, and serial printers. An example of a TTY connection is a PC with a terminal emulation program connected to the device using a dial-up modem, or a Telnet connection.

Standard logging is enabled by default, but filtering by the Embedded Syslog Manager (ESM) is disabled by default. If standard logging has been disabled on your system (using the **no logging on** command), standard logging must be reenabled using the **logging on** command before using the **logging monitor filtered** command.

ESM uses syslog filter modules, which are Tcl script files stored locally or on a remote device. The syslog filter modules must be configured using the **logging filter** command before system logging messages can be filtered.

When ESM filtering is enabled, all messages sent to the monitor have the configured syslog filter modules applied. To disable filtered logging to the monitor and return to standard logging, issue the standard **logging monitor** command (without the **filtered** keyword). To disable all logging to the monitor connections, use the **no logging monitor** command, with or without the **filtered** keyword.

Examples

In the following example, the user enables ESM filtered logging to the monitor connections:

```
Router(config)# logging filter tftp://209.165.200.225/ESM/escalate.tcl
Router(config)# logging filter slot0:/email.tcl user@example.com
Router(config)# logging monitor filtered
```

Related Commands

Command	Description
logging monitor	Enables standard system message logging to all monitor (TTY) connections.
show logging xml	Displays the state of XML-formatted system message logging, followed by the contents of the XML syslog buffer.

logging origin-id

To add an origin identifier to system logging messages sent to remote hosts, use the **logging origin-id** command in global configuration mode. To disable the origin identifier, use the **no** form of this command.

logging origin-id {**hostname** | **ip** | **string** *user-defined-id*}

no logging origin-id {**hostname** | **ip** | **string** *user-defined-id*}

Syntax Description

hostname	Specifies that the hostname will be used as the message origin identifier.
ip	Specifies that the IP address of the sending interface will be used as the message origin identifier.
string *user-defined-id*	Allows you to enter your own identifying description. The *user-defined-id* argument is a string you specify. • You can enter a string with no spaces or use delimiting quotation marks to enclose a string with spaces.

Defaults

Disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.3(1)	The **string** *user-defined-id* syntax was added.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

The origin identifier is added to the beginning of all system logging (syslog) messages sent to remote hosts. The identifier can be the hostname, the IP address, or any text that you specify. The origin identifier is not added to messages sent to local destinations (the console, monitor, or buffer).

The origin identifier is useful for identifying the source of system logging messages in cases where you send syslog output from multiple devices to a single syslog host.

When specifying your own identification string using the **logging origin-id string** *user-defined-id* command, the system expects a string without spaces. For example:

```
Router(config)# logging origin-id string Cisco_Systems
```

To uses spaces (multiple words) or additional syntax, enclose the string with quotes. For example:

```
Router(config)# logging origin-id string "Cisco Systems, Inc."
```

Examples

In the following example, the origin identifier “Domain 1, router B” will be added to the beginning of all system logging messages sent to remote hosts:

```
Router(config)# logging origin-id string "Domain 1, router B"
```

In the following example, all logging message sent to remote hosts will have the IP address configured for the Serial 1 interface added to the beginning of the message:

```
Router(config)# logging host 209.165.200.225
Router(config)# logging trap 5
Router(config)# logging source-interface serial 1
Router(config)# logging origin-id ip
```

Related Commands

Command	Description
logging host	Enables system message logging to a remote host.
logging source-interface	Forces logging messages to be sent from a specified interface, instead of any available interface.
logging trap	Configures the severity level at or numerically below which logging messages should be sent to a remote host.

show logging

To display the state of system logging (syslog) and the contents of the standard system logging buffer, use the **show logging** command in privileged EXEC mode.

show logging [**slot** *slot-number* | **summary**]

Syntax Description

slot *slot-number*	(Optional) Displays information in the syslog history table for a specific line card. Slot numbers range from 0 to 11 for the Cisco 12012 Internet router and 0 to 7 for the Cisco 12008 Internet router.
summary	(Optional) Displays counts of messages by type for each line card.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.0	This command was introduced.
11.2 GS	The **slot** and **summary** keywords were added for the Cisco 12000 family.
12.2(8)T	Command output was expanded to show the status of the logging count facility (“Count and timestamp logging messages”).
12.2(15)T	Command output was expanded to show the status of XML syslog formatting.
12.3(2)T	Command output was expanded (on supported software images) to show details about the status of system logging processed through the Embedded Syslog Manager (ESM). These lines appear as references to “filtering” or “filter modules”.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

This command displays the state of syslog error and event logging, including host addresses, and which logging destinations (console, monitor, buffer, or host) logging is enabled. This command also displays Simple Network Management Protocol (SNMP) logging configuration parameters and protocol activity.

This command will also display the contents of the standard system logging buffer, if logging to the buffer is enabled. Logging to the buffer is enabled or disabled using the [**no**] **logging buffered** command. The number of system error and debugging messages in the system logging buffer is determined by the configured size of the syslog buffer. This size of the syslog buffer is also set using the **logging buffered** command.

To enable and set the format for syslog message timestamping, use the **service timestamps log** command.

If debugging is enabled (using any **debug** command), and the logging buffer is configured to include level 7 (debugging) messages, debug output will be included in the system log. Debugging output is not formatted like system error messages and will not be preceded by the percent symbol (%).

Examples

The following is sample output from the **show logging** command on a software image that supports the Embedded Syslog Manager (ESM) feature:

```
Router# show logging

Syslog logging: enabled (10 messages dropped, 5 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 31 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: disabled
    Buffer logging: level errors, 36 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.


    Trap logging: level informational, 45 message lines logged

Log Buffer (8192 bytes):
```

Table 2 describes the significant fields shown in the display.

Table 2 show logging Field Descriptions

Field	Description
Syslog logging:	Shows general state of system logging (enabled or disabled), the status of logged messages (number of messages dropped, rate-limited, or flushed), and whether XML formatting or ESM filtering is enabled.
Console logging:	Logging to the console port. Shows "disabled" or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging console**, **logging console xml**, or **logging console filtered** commands.
Monitor logging:	Logging to the monitor (all TTY lines). Shows "disabled" or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging monitor**, **logging monitor xml**, or **logging monitor filtered** commands.
Buffer logging:	Logging to the standard syslog buffer. Shows "disabled" or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. Corresponds to the configuration of the **logging buffered**, **logging buffered xml**, or **logging buffered filtered** commands.

Table 2 show logging Field Descriptions

Field	Description
Trap logging:	Logging to a remote host (syslog collector). Shows “disabled” or, if enabled, the severity level limit, number of messages logged, and whether XML formatting or ESM filtering is enabled. (The word “trap” means a trigger in the system software for sending error messages to a remote host.) Corresponds to the configuration of the **logging host** command. The severity level limit is set using the **logging trap** command.
SNMP logging	Displays whether SNMP logging is enabled, the number of messages logged, and the retransmission interval. If not shown on your platform, use the **show logging history** command.
Logging Exception size (8192 bytes)	Corresponds to the configuration of the **logging exception** command.
Count and timestamp logging messages:	Corresponds to the configuration of the **logging count** command.
No active filter modules.	Appears if no syslog filter modules are configured with the **logging filter** command. Syslog filter modules are Tcl script files used when the Embedded Syslog Manager (ESM) is enabled. ESM is enabled when any of the **filtered** keywords are used in the logging commands. If configured, the URL and filename of configured syslog filter modules will appear at this position in the output. Syslog filter modules are executed in the order in which they appear here.
Log Buffer (8192 bytes):	The value in parentheses corresponds to the configuration of the **logging buffered** *buffer-size* command. If no messages are currently in the buffer, the output ends with this line. If messages are stored in the syslog buffer, they appear after this line.

The following example includes syslog messages from the system buffer, with timestamping. Note that in this example, the software image does not support XML formatting or ESM filtering of syslog messages.

```
Router> show logging

Syslog logging:enabled (2 messages dropped, 0 flushes, 0 overruns)
    Console logging:disabled
    Monitor logging:level debugging, 0 messages logged
    Buffer logging:level debugging, 4104 messages logged
    Trap logging:level debugging, 4119 message lines logged
        Logging to 216.231.111.14, 4119 message lines logged
Log Buffer (262144 bytes):

Jul 11 12:17:49 EDT:%BGP-4-MAXPFX:No. of prefix received from 209.165.200.225
(afi 0) reaches 24, max 24
! THE FOLLOWING LINE IS A DEBUG MESSAGE FROM NTP.
! NOTE THAT IT IS NOT PRECEEDED BY THE % SYMBOL.
Jul 11 12:17:48 EDT: NTP: Maxslew = 213866
Jul 11 15:15:41 EDT:%SYS-5-CONFIG:Configured from
tftp://host.com/addc5505-rsm.nyiix
```

```
.Jul 11 15:30:28 EDT:%BGP-5-ADJCHANGE:neighbor 209.165.200.226 Up
.Jul 11 15:31:34 EDT:%BGP-3-MAXPFXEXCEED:No. of prefix received from
209.165.200.226 (afi 0):16444 exceed limit 375
.Jul 11 15:31:34 EDT:%BGP-5-ADJCHANGE:neighbor 209.165.200.226 Down BGP
Notification sent
.Jul 11 15:31:34 EDT:%BGP-3-NOTIFICATION:sent to neighbor 209.165.200.226 3/1
(update malformed) 0 bytes
 .
 .
 .
```

The software clock keeps an “authoritative” flag that indicates whether the time is authoritative (believed to be accurate). If the software clock has been set by a timing source (for example, via NTP), the flag is set. If the time is not authoritative, it will be used only for display purposes. Until the clock is authoritative and the “authoritative” flag is set, the flag prevents peers from synchronizing to the software clock.

Table 3 describes the symbols that proceed the timestamp.

Table 3 Timestamping Symbols for syslog Messages

Symbol	Description	Example
*	Time is not authoritative: the software clock is not in sync or has never been set.	*15:29:03.158 UTC Tue Feb 25 2003:
(blank)	Time is authoritative: the software clock is in sync or has just been set manually.	15:29:03.158 UTC Tue Feb 25 2003:
.	Time is authoritative, but NTP is not synchronized: the software clock was in sync, but has since lost contact with all configured NTP servers.	.15:29:03.158 UTC Tue Feb 25 2003:

The following is sample output from the **show logging summary** command for a Cisco 12012 router. A number in the column indicates that the syslog contains that many messages for the line card. For example, line card in slot 9 has 1 error message, 4 warning messages, and 47 notification messages.

Note For similar log counting on other platforms, use the **show logging count** command.

```
Router# show logging summary

+-----+-------+-------+-------+-------+-------+-------+-------+-------+
 SLOT | EMERG | ALERT | CRIT  | ERROR |WARNING| NOTICE| INFO  | DEBUG |
+-----+-------+-------+-------+-------+-------+-------+-------+-------+
|* 0* |     . |     . |     . |     . |     . |     . |     . |     . |
|  1  |       |       |       |       |       |       |       |       |
|  2  |       |       |       |     1 |     4 |    45 |       |       |
|  3  |       |       |       |       |       |       |       |       |
|  4  |       |       |       |     5 |     4 |    54 |       |       |
|  5  |       |       |       |       |       |       |       |       |
|  6  |       |       |       |       |       |       |       |       |
|  7  |       |       |       |    17 |     4 |    48 |       |       |
|  8  |       |       |       |       |       |       |       |       |
|  9  |       |       |       |     1 |     4 |    47 |       |       |
| 10  |       |       |       |       |       |       |       |       |
| 11  |       |       |       |    12 |     4 |    65 |       |       |
+-----+-------+-------+-------+-------+-------+-------+-------+-------+
Router#
```

Table 4 describes the logging level fields shown in the display.

Table 4 show logging summary Field Descriptions

Field	Description
SLOT	Indicates the slot number of the line card. An asterisk next to the slot number indicates the GRP card whose error message counts are not displayed. For information on the GRP card, use the **show logging** command.
EMERG	Indicates that the system is unusable.
ALERT	Indicates that immediate action is needed.
CRIT	Indicates a critical condition.
ERROR	Indicates an error condition.
WARNING	Indicates a warning condition.
NOTICE	Indicates a normal but significant condition.
INFO	Indicates an informational message only.
DEBUG	Indicates a debugging message.

Related Commands

Command	Description
clear logging	Clears messages from the logging buffer.
logging count	Enables the error log count capability.
logging history size	Changes the number of syslog messages stored in the history table of the router.
logging linecard	Logs messages to an internal buffer on a line card and limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above level.
service timestamps	Configures the system to timestamp debugging or logging messages.
show logging count	Displays a summary of system error messages (syslog messages) by facility and severity.
show logging xml	Displays the state of system logging and the contents of the XML-specific logging buffer.

Glossary

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

console — In the context of this feature, specifies the connection (CTY or console line) to the console port of the router. Typically, this is a terminal attached directly to the console port, or a PC with a terminal emulation program. Corresponds to the **show terminal** command.

monitor — In the context of this feature, specifies the TTY (TeleTYpe terminal) line connection at a line port. In other words, the "monitor" keyword corresponds to a terminal line connection or a Telnet (terminal emulation) connection. TTY lines (also called ports) communicate with peripheral devices such as terminals, modems, and serial printers. An example of a TTY connection is a PC with a terminal emulation program connected to the device using a dial-up modem.

SEMs—Abbreviation for system error messages. "System error messages" is the term formerly used for messages generated by the system logging (syslog) process. Syslog messages use a standardized format, and come in 8 severity levels, from "emergencies" (level 0) to "debugging" (level 7). The term "system error message" is actually misleading, as these messages can include notifications of router activity beyond "errors" (such as informational notices).

syslog—Abbreviation for the system message logging process in Cisco IOS software. Also used to identify the messages generated, as in "syslog messages." Technically, the term "syslog" refers only to the process of logging messages to a remote host or hosts, but is commonly used to refer to all Cisco IOS system logging processes.

trap — A trigger in the system software for sending error messages. In the context of this feature, "trap logging" means logging messages to a remote host. The remote host is actually a syslog host from the perspective of the device sending the trap messages, but because the receiving device typically provides collected syslog data to other devices, the receiving device is also referred to as a "syslog server."

Appendix: Syslog Filter Module Examples

Syslog Script Modules are Tcl scripts. The following examples are provided to assist you in developing your own Syslog Script Modules.

Note These script modules are provided as examples only, and are not supported by Cisco Systems, Inc. No guarantees, expressed or implied, are provided for the functionality or impact of these scripts.

This appendix contains the following syslog filter module examples:

- Severity Escalation Example, page 35
- Message Counting Example, page 35
- XML Tagging Example, page 39
- SMTP-based Email Alert Example, page 41
- Stream Example, page 43

Severity Escalation Example

This ESM syslog filter module example watches for a single mnemonic (supplied via the first CLI argument) and escalates the severity of the message to that specified by the second CLI argument.

```
#  ====================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Severity Escalation Filter                       ||||      ||||
#                                               ..:||||||:..:||||||:..
#                                              -------------------------
#                                              C i s c o   S y s t e m s
#  =====================================================================
#
# Usage: Set CLI Args to "mnemonic new_severity"
#
# Namespace: global

# Check for null message

if { [string length $::orig_msg] == 0} {
   return ""
}

if { [info exists ::cli_args] } {
    set args [split $::cli_args]
    if { [ string compare -nocase [lindex $args 0] $::mnemonic ] == 0 } {
        set ::severity [lindex $args 1]
        set sev_index [ string first [lindex $args 0] $::orig_msg ]
        if {  $sev_index >= 2 } {
           incr sev_index -2
           return [string replace $::orig_msg $sev_index $sev_index \
              [lindex $args 1]]
        }
    }
}
return $::orig_msg
```

Message Counting Example

This ESM syslog filter module example is divided into two files for readability. The first file allows the user to configure those messages that they wish to count and how often to summarize (correlation window) by populating the msg_to_watch array. The actual procedures are in the counting_guts.tcl file. Note the use of the separate namespace “counting” to avoid conflict with other ESM filters that may also perform background processing.

```
#  ====================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Message Counting Filter                          ||||      ||||
#                                               ..:||||||:..:||||||:..
#                                              -------------------------
#                                              C i s c o   S y s t e m s
#  ====================================================================
#
# Usage:
# 1) Define the location for the counting_guts.tcl script
#
# 2) Define message categories to count and how often to dump them (sec)
#    by populating the "msg_to_watch" array below.
#    Here we define category as facility-severity-mnemonic
#    Change dump time to 0 to disable counting for that category
#
# Namespace: counting


namespace eval ::counting {

    set sub_script_url tftp://123.123.123.123/ESM/counting_guts.tcl

    array set msg_to_watch {
        SYS-5-CONFIG_I            5
    }

#  ======================= End User Setup ==============================

# Initialize processes for counting

    if { [info exists init] == 0 } {
       source $sub_script_url
       set position $module_position
    }

# Process the message

process_category

} ;# end namespace counting
```

Message Counting Support Module (counting_guts.tcl)

```
#  ===================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Message Counting Support Module                  ||||      ||||
#                                               ..:||||||:..:||||||:..
#  (No User Modification)                      ------------------------
#                                              C i s c o  S y s t e m s
#  ====================================================================


namespace eval ::counting {

# namespace variables

array set cat_msg_sev {}
array set cat_msg_traceback {}
array set cat_msg_pid {}
array set cat_msg_proc {}
array set cat_msg_ts {}
array set cat_msg_buginfseq {}
array set cat_msg_name {}
array set cat_msg_fac {}
array set cat_msg_format {}
array set cat_msg_args {}
array set cat_msg_count {}
array set cat_msg_dump_ts {}

# Should I count this message ?

   proc query_category {cat} {
        variable msg_to_watch
        if { [info exists msg_to_watch($cat)] } {
            return $msg_to_watch($cat)
        } else {
            return 0
        }
    }

   proc clear_category {index} {
        variable cat_msg_sev
        variable cat_msg_traceback
        variable cat_msg_pid
        variable cat_msg_proc
        variable cat_msg_ts
        variable cat_msg_buginfseq
        variable cat_msg_name
        variable cat_msg_fac
        variable cat_msg_format
        variable cat_msg_args
        variable cat_msg_count
        variable cat_msg_dump_ts

        unset cat_msg_sev($index) cat_msg_traceback($index) cat_msg_pid($index)\
              cat_msg_proc($index) cat_msg_ts($index) \
              cat_msg_buginfseq($index)  cat_msg_name($index) \
              cat_msg_fac($index) cat_msg_format($index) cat_msg_args($index)\
              cat_msg_count($index) cat_msg_dump_ts($index)
    }

# send out the counted messages

   proc dump_category {category} {
        variable cat_msg_sev
        variable cat_msg_traceback
        variable cat_msg_pid
        variable cat_msg_proc
        variable cat_msg_ts
        variable cat_msg_buginfseq
        variable cat_msg_name
        variable cat_msg_fac
        variable cat_msg_format
        variable cat_msg_args
        variable cat_msg_count
        variable cat_msg_dump_ts
        variable poll_interval
        set dump_timestamp [cisco_service_timestamp]

foreach index [array names cat_msg_count $category] {
      set fsm "$cat_msg_fac($index)-$cat_msg_sev($index)-$cat_msg_name($index)"
      set ::orig_msg \
        [format "%s%s: %%%s: %s %s %s %s - (%d occurence(s) between %s and %s)"\
        $cat_msg_buginfseq($index)\
          $dump_timestamp\
          $fsm \
          [uplevel 1 [linsert $cat_msg_args($index) 0 ::format
$cat_msg_format($index) ]] \
          $cat_msg_pid($index) \
          $cat_msg_proc($index) \
          $cat_msg_traceback($index) \
          $cat_msg_count($index) \
          $cat_msg_ts($index) \
          $dump_timestamp]

# Prepare for remaining ESM filters

                set ::severity $cat_msg_sev($index)
                set ::traceback $cat_msg_traceback($index)
                set ::pid $cat_msg_pid($index)
                set ::process $cat_msg_proc($index)
                set ::timestamp $cat_msg_ts($index)
                set ::buginfseq $cat_msg_buginfseq($index)
                set ::mnemonic $cat_msg_name($index)
                set ::facility $cat_msg_fac($index)
                set ::format_string $cat_msg_format($index)
                set ::msg_args [split $cat_msg_args($index)]

                esm_errmsg $counting::position
                clear_category $index
        }
  }

# See if this message already has come through since the last dump.
# If so, increment the count, otherwise store it.

   proc process_category {} {
        variable cat_msg_sev
        variable cat_msg_traceback
        variable cat_msg_pid
        variable cat_msg_proc
        variable cat_msg_ts
        variable cat_msg_buginfseq
        variable cat_msg_name
        variable cat_msg_fac
        variable cat_msg_format
        variable cat_msg_args
        variable cat_msg_count
        variable cat_msg_dump_ts

        if { [string length $::orig_msg] == 0} {
           return ""
        }
        set category "$::facility-$::severity-$::mnemonic"

        set correlation_window [expr [ query_category $category ] * 1000]

        if { $correlation_window == 0 } {
            return $::orig_msg
        }

        set message_args [join $::msg_args]
        set index "$category,[lindex $::msg_args 0]"
        if { [info exists cat_msg_count($index)] } {
                 incr cat_msg_count($index)
         } else {
              set cat_msg_sev($index) $::severity
              set cat_msg_traceback($index) $::traceback
              set cat_msg_pid($index) $::pid
              set cat_msg_proc($index) $::process
              set cat_msg_ts($index) $::timestamp
              set cat_msg_buginfseq($index) $::buginfseq
              set cat_msg_name($index) $::mnemonic
              set cat_msg_fac($index) $::facility
              set cat_msg_format($index) $::format_string
              set cat_msg_args($index) $message_args
              set cat_msg_count($index) 1
              set cat_msg_dump_ts($index) [clock seconds]
              catch [after $correlation_window counting::dump_category $index]
         }
         return ""
   }


# Initialized
set init 1

} ;#end namespace counting
```

XML Tagging Example

This ESM syslog filter module applies user defined XML tags to syslog messages.

```
#  ===================================================================
#  Embedded Syslog Manager                                   ||        ||
#                                                            ||        ||
#  XML Tagging Filter                                       ||||      ||||
#                                                       ..:||||||:..:||||||:..
#                                                     ------------------------
#                                                     C i s c o  S y s t e m s
#  ===================================================================
#
# Usage: Define desired tags below.
#
# Namespace: xml

# Check for null message
```

```
    if { [string length $::orig_msg] == 0} {
       return ""
    }

namespace eval xml {

####  define tags ####
set MSG_OPEN "<ios-log-msg>"
set MSG_CLOSE "</ios-log-msg>"
set FAC_OPEN   "<facility>"
set FAC_CLOSE  "</facility>"
set SEV_OPEN   "<severity>"
set SEV_CLOSE  "</severity>"
set MNE_OPEN   "<msg-id>"
set MNE_CLOSE  "</msg-id>"
set SEQ_OPEN   "<seq>"
set SEQ_CLOSE  "</seq>"
set TIME_OPEN  "<time>"
set TIME_CLOSE "</time>"
set ARGS_OPEN  "<args>"
set ARGS_CLOSE "</args>"
set ARG_ID_OPEN "<arg id="
set ARG_ID_CLOSE "</arg>"
set PROC_OPEN "<proc>"
set PROC_CLOSE "</proc>"
set PID_OPEN "<pid>"
set PID_CLOSE "</pid>"
set TRACE_OPEN "<trace>"
set TRACE_CLOSE "</trace>"

#  ====================== End User Setup ==============================

#### clear result ####

set result ""

#### message opening, facility, severity, and name ####
append result $MSG_OPEN $FAC_OPEN $::facility $FAC_CLOSE $SEV_OPEN $::severity
$SEV_CLOSE $MNE_OPEN $::mnemonic $MNE_CLOSE

#### buginf sequence numbers ####

if { [string length $::buginfseq ] > 0 } {
    append result $SEQ_OPEN $::buginfseq $SEQ_CLOSE
}

#### timestamps ####

if { [string length $::timestamp ] > 0 } {
    append result $TIME_OPEN $::timestamp $TIME_CLOSE
}

#### message args ####
if { [info exists ::msg_args] } {
    if { [llength ::msg_args] > 0 } {
        set i 0
        append result $ARGS_OPEN
        foreach arg $::msg_args {
            append result $ARG_ID_OPEN $i ">" $arg $ARG_ID_CLOSE
                incr i
        }
        append result $ARGS_CLOSE
    }
}

#### traceback ####

if { [string length $::traceback ] > 0 } {
    append result $TRACE_OPEN $::traceback $TRACE_CLOSE
}

#### process ####

if { [string length $::process ] > 0 } {
    append result $PROC_OPEN $::process $PROC_CLOSE
}

#### pid ####

if { [string length $::pid ] > 0 } {
    append result $PID_OPEN $::pid $PID_CLOSE
}

#### message close ####

append result $MSG_CLOSE


return "$result"

} ;# end namespace xml
```

SMTP-based Email Alert Example

This ESM syslog filter module example watches for configuration messages and sends them to the email address supplied as a CLI argument. This filter is divided into two files. The first file implements the filter, and the second file implements the SMTP client.

```
#  ===================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Email Filter                                     ||||      ||||
#  (Configuration Change Warning)              ..:||||||:..:||||||:..
#                                             ------------------------
#                                              C i s c o  S y s t e m s
#   ===================================================================

# Usage:  Provide email address as CLI argument.  Set email server IP in
#         email_guts.tcl
#
# Namespace: email
```

```
if { [info exists email::init] == 0 } {
   source tftp://123.123.123.123/ESM/email_guts.tcl
}

# Check for null message

if { [string length $::orig_msg] == 0} {
      return ""
   }

if { [info exists ::msg_args] } {
    if { [string compare -nocase CONFIG_I $::mnemonic ] == 0 } {
                email::sendmessage $::cli_args $::mnemonic \
                [string trim $::orig_msg]
    }
}
return $::orig_msg
```

Email Support Module (email_guts.tcl)

```
#  ===================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Email Support Module                             ||||      ||||
#                                               ..:||||||:..:||||||:..
#                                             ------------------------
#                                              C i s c o  S y s t e m s
#   ===================================================================
#
# Usage: Set email host IP, from, and friendly strings below.
#

namespace eval email {

    set sendmail(smtphost) 64.102.17.214
    set sendmail(from) $::hostname
    set sendmail(friendly) $::hostname


    proc sendmessage {toList subject body} {

        variable sendmail

        set smtphost $sendmail(smtphost)
        set from $sendmail(from)
        set friendly $sendmail(friendly)

        set sockid [socket $smtphost 25]

## DEBUG
set status [catch {
        puts $sockid "HELO $smtphost"
        flush $sockid
        set result [gets $sockid]
        puts $sockid "MAIL From:<$from>"
        flush $sockid
        set result [gets $sockid]

        foreach to $toList {
            puts $sockid "RCPT To:<$to>"
            flush $sockid
        }

        set result [gets $sockid]

        puts  $sockid "DATA "
        flush $sockid
        set result [gets  $sockid]

        puts  $sockid "From: $friendly <$from>"
        foreach to $toList {
            puts $sockid "To:<$to>"
        }
        puts  $sockid "Subject: $subject"
        puts  $sockid "\n"

        foreach line [split $body "\n"] {
            puts  $sockid " $line"
        }

        puts  $sockid "."
        puts  $sockid "QUIT"
        flush $sockid
        set result [gets  $sockid]

} result]

        catch {close $sockid }
    if {$status} then {
        return -code error $result
    }
}

} ;# end namespace email

set email::init 1
```

Stream Example

This ESM syslog filter module example watches for a given facility (first CLI argument) and routes these messages to a given stream (second CLI argument).

```
#  ===================================================================
#  Embedded Syslog Manager                           ||        ||
#                                                    ||        ||
#  Stream Filter (Facility)                         ||||      ||||
#                                               ..:||||||:..:||||||:..
#                                             ------------------------
#                                             C i s c o  S y s t e m s
#   ===================================================================

# Usage:  Provide facility and stream as CLI arguments.
#
# Namespace: global

# Check for null message

#  ======================= End User Setup ==============================

set args [split $::cli_args]

if { [info exists ::msg_args] } {
    if { $::facility == [lindex $args 0] } {
                set ::stream [lindex $args 1]
    }
}
return $::orig_msg}
```

Source IP Tagging Example

The **logging source-interface** CLI command can be used to specify a source IP address in all syslog packets sent from the router. The following syslog filter module example demonstrates the use of **show** CLI commands (**show running-config** and **show ip interface** in this case) within a filter module to add the source IP address to syslog messages. The script looks for the local namespace variable "source_ip::init" first. If the variable is not defined in the first syslog message processed, the filter will run the **show** commands and use regular expressions to get the source-interface and then its IP address.

Note that in this script, the **show** commands are only run once. If the source-interface or its IP address were to be changed, the filter would have to be re-initialized to pick up the new information. (You could have the show commands run on every syslog message, but this would not scale very well.)

```
#  ===================================================================
#  Embedded Syslog Manager                            ||        ||
#                                                     ||        ||
#  Source IP Module                                  ||||      ||||
#                                                ..:||||||:..:||||||:..
#                                              ------------------------
#                                              C i s c o  S y s t e m s
#   ===================================================================

# Usage: Adds Logging Source Interface IP address to all messages.
#
# Namespace:source_ip
#
#  ======================= End User Setup ==============================

namespace eval ::source_ip {

    if { [info exists init] == 0 } {
       if { [catch {regexp {^logging source-interface (.*$)} [exec show
run | inc logging source-interface] match source_int}]} {
           set suffix "No source interface specified"
       } elseif { [catch {regexp {Internet address is (.*)/.*$} [exec
show ip int $source_int | inc Internet] match ip_addr}]} {
           set suffix "No IP address configured for source interface"
       } else {
           set suffix $ip_addr
       }
       set init 1
    }

    if { [string length $::orig_msg] == 0} {
      return ""
    }

    return "$::orig_msg - $suffix"

} ;# end namespace source_ip
```