Monitoring Alarms and Alarm Log Correlation

This module describes the concepts and tasks related to monitoring or displaying router alarms, configuring alarm log correlation, monitoring alarm logs, and correlated event records. Alarm log correlation extends system logging to include the ability to group and filter messages generated by various applications and system servers and to isolate root messages on the router.

This module describes the new and revised tasks you need to perform to implement logging correlation and monitor alarms on your network.

Note For more information about system logging on Cisco IOS XR Software and complete descriptions of the alarm management and logging correlation commands listed in this module, see the Related Documents, on page 45 section of this module.

Feature History for Monitoring Alarms and Implementing Alarm Log Correlation

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.8.0	SNMP alarm correlation feature was added.
Release 6.4.1	The Pulse Queue State Monitoring feature was introduced in Cisco IOS XR 32 bit operating system.

- Prerequisites for Implementing Alarm Log Correlation, on page 2
- Information About Monitoring Alarms and Implementing Alarm Log Correlation, on page 2
- How to Implement and Monitor Alarm Management and Logging Correlation, on page 13
- Configuration Examples for Alarm Management and Logging Correlation, on page 41
- Pulse Queue State Monitoring, on page 44
- Additional References, on page 45

Prerequisites for Implementing Alarm Log Correlation

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Monitoring Alarms and Implementing Alarm Log Correlation

Displaying Router Alarms

You can view the router alarms in brief and detail.

Execute the command **show alarms brief** to view the router alarms in brief.

```
RP/0/RSP0/CPU0:router#show alarms brief

------------------------------------------------------------------------
Active Alarms for 1/0
------------------------------------------------------------------------
Location    Severity   Group     Set time               Description

------------------------------------------------------------------------
0/1/CPU0   Critical   Fabric   11/11/2022 10:34:22 IST  LC Bandwidth Insufficient To Support
 Line Rate Traffic
1/0/CPU0    Major    Software  11/11/2022 10:43:36 IST   Optics1/0/0/20 - hw_optics:  RX
LOS LANE-0 ALARM
1/0/CPU0    Major    Software  11/11/2022 10:43:36 IST   Optics1/0/0/20 - hw_optics:  RX
LOS LANE-1 ALARM
------------------------------------------------------------------------------------------
History Alarms for 1/0
------------------------------------------------------------------------------------------
No entries.

------------------------------------------------------------------------------------------
Suppressed Alarms for 1/0
------------------------------------------------------------------------------------------
No entries.

------------------------------------------------------------------------------------------
Conditions for 1/0
------------------------------------------------------------------------------------------
No entries.
```

Execute the command **show alarms detail** to view the router alarms in detail.

```
RP/0/RSP0/CPU0:ddc2-uut#show alarms detail

-------------------------------------------------------
Active Alarms for 1/0
-------------------------------------------------------
Description:            LC Bandwidth Insufficient To Support Line Rate Traffic
```

```
Location:                1/0/CPU0

AID:                     XR_FABRIC/SW_MISC_ERR/18

Tag String:              FAM_FAULT_TAG_HW_FIA_LC_BANDWIDTH

Module Name:             N/A

EID:                     MODULE/MSC/1:MODULE/SLICE/1:MODULE/PSE/1

Reporting Agent ID:      524365
Pending Sync:            false
Severity:                Critical
Status:                  Set
Group:                   Fabric
Set Time:                11/16/2022 20:44:44 IST
Clear Time:              -
Service Affecting:       NotServiceAffecting
Transport Direction:     NotSpecified
Transport Source:        NotSpecified
Interface:               N/A

Alarm Name:              LC-BW-DEG
-------------------------------------------------------------

History Alarms for 1/0
-------------------------------------------------------------
No entries.

-------------------------------------------------------------
Suppressed Alarms for 1/0
-------------------------------------------------------------
No entries.

-------------------------------------------------------------
Conditions for 1/0
-------------------------------------------------------------
No entries.

-------------------------------------------------------------
Clients for 1/0
-------------------------------------------------------------
Agent Name:              optics_fm.xml

Agent ID:                196678
Agent Location:          1/0/CPU0

Agent Handle:            93827323237168

Agent State:             Registered
Agent Type:              Producer
Agent Filter Display:    false
Agent Subscriber ID:     0
Agent Filter Severity:   Unknown
Agent Filter State:      Unknown
Agent Filter Group:      Unknown
Agent Connect Count:     1
Agent Connect Timestamp: 11/16/2022 20:40:18 IST
Agent Get Count:         0
Agent Subscribe Count:   0
Agent Report Count:      8
-------------------------------------------------------------
Statistics for 1/0
-------------------------------------------------------------
```

```
Alarms Reported:                9
Alarms Dropped:                 0
Active (bi-state set):          9
History (bi-state cleared):     0
Suppressed:                     0
Dropped Invalid AID:            0
Dropped No Memory:              0
Dropped DB Error:               0
Dropped Clear Without Set:      0
Dropped Duplicate:              0
Cache Hit:                      0
Cache Miss:                     0
```

Alarm Logging and Debugging Event Management System

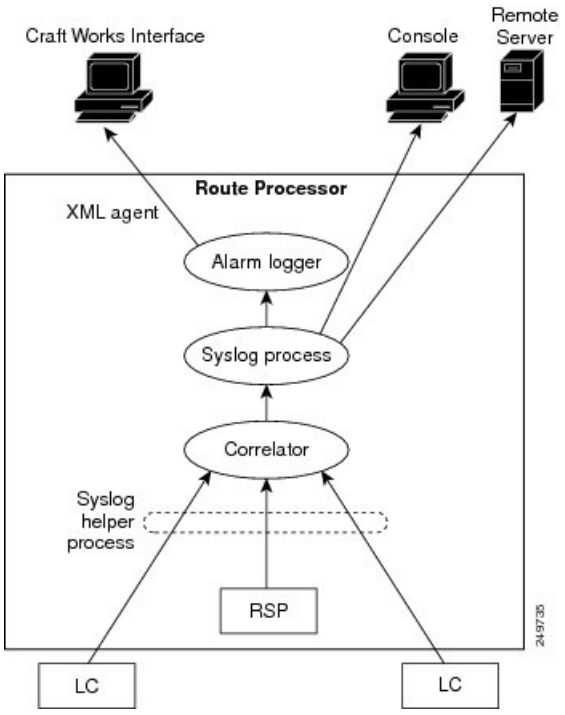
Cisco IOS XR Software Alarm Logging and Debugging Event Management System (ALDEMS) is used to monitor and store alarm messages that are forwarded by system servers and applications. In addition, ALDEMS correlates alarm messages forwarded due to a single root cause.

ALDEMS enlarges on the basic logging and monitoring functionality of Cisco IOS XR Software, providing the level of alarm and event management necessary for a highly distributed system .

Cisco IOS XR Software achieves this necessary level of alarm and event management by distributing logging applications across the nodes on the system.

Figure 1: ALDEMS Component Communications, on page 4 illustrates the relationship between the components that constitute ALDEMS.

Figure 1: ALDEMS Component Communications



Correlator

The correlator receives messages from system logging (syslog) helper processes that are distributed across the nodes on the router and forwards syslog messages to the syslog process. If a logging correlation rule is configured, the correlator captures messages searching for a match with any message specified in the rule. If the correlator finds a match, it starts a timer that corresponds to the timeout interval specified in the rule. The correlator continues searching for a match to messages in the rule until the timer expires. If the root case message was received, then a correlation occurs; otherwise, all captured messages are forwarded to the syslog. When a correlation occurs, the correlated messages are stored in the logging correlation buffer. The correlator tags each set of correlated messages with a correlation ID.

Note For more information about logging correlation, see the Logging Correlation, on page 8 section.

System Logging Process

By default, routers are configured to send system logging messages to a system logging (syslog) process. Syslog messages are gathered by syslog helper processes that are distributed across the nodes on the system. The system logging process controls the distribution of logging messages to the various destinations, such as the system logging buffer, the console, terminal lines, or a syslog server, depending on the network device configuration.

Bias-Free Terminology in Syslogs

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Bias-Free Terminology in Syslogs	Release 7.6.1	As part of ongoing efforts to use inclusive language in the IOS XR software and documentation, Cisco has created the **service cli neutral enable** command. When you execute this command, biased terms found in the logs of certain features are replaced with bias-free terms. There are certain other features that are not controlled by the command. The biased terms in these features are automatically replaced with bias-free terms. Cisco expects to convert these features to be controlled by the command in subsequent releases.

You have various features deployed in your network. The feature-specific log that you retrieve from the syslog server may contain biased terms. As part of the ongoing efforts to remove such biased terms, Cisco has introduced the **service cli neutral enable** command. Use this command to replace biased terms with bias-free terms in the logs of certain features. There are certain other features that automatically reflect these changes, and Cisco expects to convert these features to be controlled by the CLI in subsequent releases.

This table lists the features for which Cisco has automatically replaced biased terms. The table also lists the replaced, bias-free term.

Existing Log Information	Modified Log Information
WARN_TRAFFIC_**SLAVE**	WARN_TRAFFIC_**SUBORDINATE**
VRRP control packet received from %s on MGO **slave**: %s VRID %u	VRRP control packet received from %s on MGO **subordinate**: %s VRID %u
Type6 **masterkey** update still in progress. masterkey get failed	Type6 **key** update still in progress. Type6 key get failed.
Mastered get failed because of Type6 **masterkey** update still in progress.	Type6 **keys get** failed because of Type6 key update still in progress.
Received **master key** update event. Migrating existing type6 passwords. Database will be updated.	Received type6 **key** update event. Migrating existing type6 passwords. Database will be updated.
Completed migration of existing type6 passwords using the new **masterkey**. Database successfully updated.	Completed migration of existing type6 passwords using the new type6 **key**. Database successfully updated.
Decryption failed for keychain %s, keyid %s type6 password. Failed to migrate type6 keychains with the new **masterkey**	Decryption failed for keychain %s, keyid %s type6 password. Failed to migrate type6 keychains with the new type6 **key**
Keychain failed to process the **masterkey** update abort event from type6, err : %s	Keychain failed to process the type6 **key** update abort event from type6, err : %s
Type6 **masterkey** update : %s	Type6 **key** update : %s
Type6 **masterkey** %s successfully	Type6 **key** %s successfully
Type6 **masterkey** %s timeout	Type6 **key** %s timeout
Type6 **masterkey** %s successfully	Type6 **key** %s successfully
Type6 **masterkey** %s failed with Error: %s	Type6 **key** %s failed with Error: %s
All clients have not completed the **masterkey** abort operation. Forcing the abort operation.	All clients have not completed the type6 **key** update abort operation. Forcing the abort operation.
Type6 **masterkey** reconfig is inprogress. Unregister of client isn't allowed.	Type6 **key** reconfig is inprogress. Unregister of client is not allowed.
MASTERKEY_UPDATE	**MK**_UPDATE
WARN_TRAFFIC_**SLAVE**	WARN_TRAFFIC_**SUBORDINATE**
HSRP control packet received from %s on MGO **slave** group: %s group %u	HSRP control packet received from %s on MGO **subordinate** group: %s group %u

Existing Log Information	Modified Log Information
ROLE_**MASTER**	ROLE_ **PRIMARY**
ROLE_**SLAVE**	ROLE_**BACKUP**
Message: SYNC_DONE "please check if the affected sessions have been flagged as failed and these would be retried on the **Slave** side."	Message: SYNC_DONE "please check if the affected sessions have been flagged as failed and these would be retried on the **Backup** side."
SLAVE_UPD_FAIL	**BACKUP**_UPD_FAIL

This table lists the features for which you can run the **service cli neutral enable** command to change the biased terms.

Existing Log Information	Modified Log Information
ROLE_**MASTER**	"ROLE_**PRIMARY**"
ROLE_**SLAVE**	"ROLE_**BACKUP**"
SLAVE_UPD_FAIL	"**BACKUP**_UPD_FAIL"

Enable Bias-Free Configuration

Configuration Example

To replace biased terms with bias-free terms in your logs, run the following command:

```
RP/0/RP0/CPU0:router(config)#service cli neutral enable
RP/0/RP0/CPU0:router(config)#commit
Tue Mar 15 08:11:10.253 UTC
RP/0/RP0/CPU0:router(config)#end
```

If required, use the **no** form of the command to recall biased terms in your logs.

Verification

To verify that biased terms are replaced, run the following command

```
RP/0/RP0/CPU0:router#show running-config service cli neutral enable
Tue Mar 15 08:11:21.893 UTC
service cli neutral enable
```

This codeblock shows a sample output when the CLI is enabled:

```
RP/0/0/CPU0:Apr 13 12:01:04.371 IST: serg_agt[1188]: %INFRA-SERG-5-ROLE_PRIMARY : Session
Redundancy role change to Primary from Backup for group 2 reason ADMIN
LC/0/1/CPU0:Apr 13 12:01:04.372 IST: serg_agt[323]: %INFRA-SERG-5-ROLE_BACKUP : Session
Redundancy role change to Backup from Primary for group 1 reason ADMIN
RP/0/0/CPU0:Apr 13 12:01:04.377 IST: serg_agt[1188]: %INFRA-SERG-5-ROLE_BACKUP : Session
Redundancy role change to Backup from Primary for group 2 reason ADMIN
LC/0/1/CPU0:Apr 13 12:01:04.377 IST: serg_agt[323]: %INFRA-SERG-5-ROLE_PRIMARY : Session
Redundancy role change to Primary from Backup for group 1 reason ADMIN
```

This codeblock shows a sample output when the CLI is not enabled:

```
RP/0/0/CPU0:Apr 13 11:50:03.333 IST: serg_agt[1188]: %INFRA-SERG-5-ROLE_SLAVE : Session
Redundancy role change to Slave from None for group 2 reason PEER-UP
```

```
LC/0/1/CPU0:Apr 13 11:50:03.334 IST: serg_agt[323]: %INFRA-SERG-5-ROLE_MASTER : Session
Redundancy role change to Master from None for group 1 reason PEER-UP
RP/0/0/CPU0:Apr 13 11:50:03.338 IST: serg_agt[1188]: %INFRA-SERG-5-ROLE_MASTER : Session
Redundancy role change to Master from None for group 2 reason PEER-UP
LC/0/1/CPU0:Apr 13 11:50:03.339 IST: serg_agt[323]: %INFRA-SERG-5-ROLE_SLAVE : Session
Redundancy role change to Slave from None for group 1 reason PEER-UP
```

Alarm Logger

The alarm logger is the final destination for system logging messages forwarded on the router. The alarm logger stores alarm messages in the logging events buffer. The logging events buffer is circular; that is, when full, it overwrites the oldest messages in the buffer.

Note Alarms are prioritized in the logging events buffer. When it is necessary to overwrite an alarm record, the logging events buffer overwrites messages in the following order: nonbistate alarms first, then bistate alarms in the CLEAR state, and, finally, bistate alarms in the SET state. For more information about bistate alarms, see the Bistate Alarms, on page 10 section.

When the table becomes full of messages caused by bistate alarms in the SET state, the earliest bistate message (based on the message time stamp, not arrival time) is reclaimed before others. The buffer size for the logging events buffer and the logging correlation buffer, thus, should be adjusted so that memory consumption is within your requirements.

A table-full alarm is generated each time the logging events buffer wraps around. A threshold crossing notification is generated each time the logging events buffer reaches the capacity threshold.

Messages stored in the logging events buffer can be queried by clients to locate records matching specific criteria. The alarm logging mechanism assigns a sequential, unique ID to each alarm message.

Logging Correlation

Logging correlation can be used to isolate the most significant root messages for events affecting system performance. For example, the original message describing a card online insertion and removal (OIR) of a card can be isolated so that only the root-cause message is displayed and all subsequent messages related to the same event are correlated. When correlation rules are configured, a common root event that is generating secondary (non-root-cause) messages can be isolated and sent to the syslog, while secondary messages are suppressed. An operator can retrieve all correlated messages from the logging correlator buffer to view correlation events that have occurred.

Correlation Rules

Correlation rules can be configured to isolate root messages that may generate system alarms. Correlation rules prevent unnecessary stress on ALDEMS caused by the accumulation of unnecessary messages. Each correlation rule hinges on a message identification, consisting of a message category, message group name, and message code. The correlator process scans messages for occurrences of the message.

If the correlator receives a root message, the correlator stores it in the logging correlator buffer and forwards it to the syslog process on the RP. From there, the syslog process forwards the root message to the alarm logger in which it is stored in the logging events buffer. From the syslog process, the root message may also be forwarded to destinations such as the console, remote terminals, remote servers, the fault management system, and the Simple Network Management Protocol (SNMP) agent, depending on the network device

configuration. Subsequent messages meeting the same criteria (including another occurrence of the root message) are stored in the logging correlation buffer and are forwarded to the syslog process on the router.

If a message matches multiple correlation rules, all matching rules apply and the message becomes a part of all matching correlation queues in the logging correlator buffer.

The following message fields are used to define a message in a logging correlation rule:

- Message category
- Message group
- Message code

Wildcards can be used for any of the message fields to cover wider set of messages. Configure the appropriate set of messages in a logging correlation rule configuration to achieve correlation with a narrow or wide scope (depending on your objective).

Types of Correlation

There are two types of correlation that are configured in rules to isolate root-cause messages:

Nonstateful Correlation—This correlation is fixed after it has occurred, and non-root-cause alarms that are suppressed are never forwarded to the syslog process. All non-root-cause alarms remain buffered in correlation buffers.

Stateful Correlation—This correlation can change after it has occurred, if the bistate root-cause alarm clears. When the alarm clears, all the correlated non-root-cause alarms are sent to syslog and are removed from the correlation buffer. Stateful correlations are useful to detect non-root-cause conditions that continue to exist even if the suspected root cause no longer exists.

Application of Rules and Rule Sets

If a correlation rule is applied to the entire router, then correlation takes place only for those messages that match the configured cause values for the rule, regardless of the context or location setting of that message.

If a correlation rule is applied to a specific set of contexts or locations, then correlation takes place only for those messages that match the configured cause values for the rule and that match at least one of those contexts or locations.

In the case of a rule-set application, the behavior is the same; however, the apply configuration takes place for all rules that are part of the given rule set.

The **show logging correlator rule** command is used to display apply settings for a given rule, including those settings that have been configured with the **logging correlator apply ruleset** command.

Root Message and Correlated Messages

When a correlation rule is configured and applied, the correlator starts searching for a message match as specified in the rule. After a match is found, the correlator starts a timer corresponding to the timeout interval that is also specified in the rule. A message search for a match continues until the timer expires. Correlation occurs after the root-cause message is received.

The first message (with category, group, and code triplet) configured in a correlation rule defines the root-cause message. A root-cause message is always forwarded to the syslog process. See the Correlation Rules, on page 8 section to learn how the root-cause message is forwarded and stored.

Alarm Severity Level and Filtering

Filter settings can be used to display information based on severity level. The alarm filter display indicates the severity level settings used to report alarms, the number of records, and the current and maximum log size.

Alarms can be filtered according to the severity level shown in this table.

Table 2: Alarm Severity Levels for Event Logging

Severity Level	System Condition
0	Emergencies
1	Alerts
2	Critical
3	Errors
4	Warnings
5	Notifications
6	Informational

Bistate Alarms

Bistate alarms are generated by state changes associated with system hardware, such as a change of interface state from active to inactive, the online insertion and removal (OIR) of a card , or a change in component temperature. Bistate alarm events are reported to the logging events buffer by default; informational and debug messages are not.

Cisco IOS XR Software software provides the ability to reset and clear alarms. Clients interested in monitoring alarms in the system can register with the alarm logging mechanism to receive asynchronous notifications when a monitored alarm changes state.

Bistate alarm notifications provide the following information:

- The alarm state, which may be in the set state or the clear state.

Capacity Threshold Setting for Alarms

The capacity threshold setting determines when the alarm system begins reporting threshold crossing alarms. The capacity threshold for generating warning alarms is generally set at 80 percent of buffer capacity, but individual configurations may require different settings.

Hierarchical Correlation

Hierarchical correlation takes effect when the following conditions are true:

- When a single alarm is both a root cause for one rule and a non-root cause for another rule.
- When alarms are generated that result in successful correlations associated with both rules.

The following example illustrates two hierarchical correlation rules:

Rule 1	**Category**	**Group**	**Code**
Root Cause 1	Cat 1	Group 1	Code 1
Non-root Cause 2	Cat 2	Group 2	Code 2
Rule 2			
Root Cause 2	Cat 2	Group 2	Code 2
Non-root Cause 3	Cat 3	Group 3	Code 3

If three alarms are generated for Cause 1, 2, and 3, with all alarms arriving within their respective correlation timeout periods, then the hierarchical correlation appears like this:

Cause 1 -> Cause 2 -> Cause 3

The correlation buffers show two separate correlations: one for Cause 1 and Cause 2 and the second for Cause 2 and Cause 3. However, the hierarchical relationship is implicitly defined.

Note Stateful behavior, such as reparenting and reissuing of alarms, is supported for rules that are defined as stateful; that is, correlations that can change.

Context Correlation Flag

The context correlation flag allows correlations to take place on a “per context” basis or not.

This flag causes behavior change only if the rule is applied to one or more contexts. It does not go into effect if the rule is applied to the entire router or location nodes.

The following is a scenario of context correlation behavior:

- Rule 1 has a root cause A and an associated non-root cause.
- Context correlation flag is not set on Rule 1.
- Rule 1 is applied to contexts 1 and 2.

If the context correlation flag is not set on Rule 1, a scenario in which alarm A generated from context 1 and alarm B generated from context 2 results in the rule applying to both contexts regardless of the type of context.

If the context correlation flag is now set on Rule 1 and the same alarms are generated, they are not correlated as they are from different contexts.

With the flag set, the correlator analyzes alarms against the rule only if alarms arrive from the same context. In other words, if alarm A is generated from context 1 and alarm B is generated from context 2, then a correlation does not occur.

Duration Timeout Flags

The root-cause timeout (if specified) is the alternative rule timeout to use in the situation in which a non-root-cause alarm arrives before a root-cause alarm in the given rule. It is typically used to give a shorter timeout in a situation under the assumption that it is less likely that the root-cause alarm arrives, and, therefore, releases the hold on the non-root-cause alarms sooner.

Reparent Flag

The reparent flag specifies what happens to non-root-cause alarms in a hierarchical correlation when their immediate root cause clears.

The following example illustrates context correlation behavior:

- Rule 1 has a root cause A and an associated non-root cause B
- Context correlation flag is not set on Rule 1
- Rule 1 is applied to contexts 1 and 2

In this scenario, if alarm A arrives generated from context 1 and alarm B generated from context 2, then a correlation occurs—regardless of context.

If the context correlation flag is now set on Rule 1 and the same alarms are generated, they are not correlated, because they are from different contexts.

Reissue Nonbistate Flag

The reissue nonbistate flag controls whether nonbistate alarms (events) are forwarded from the correlator log if their parent bistate root-cause alarm clears. Active bistate non-root-causes are always forwarded in this situation, because the condition is still present.

The reissue-nonbistate flag allows you to control whether non-bistate alarms are forwarded.

Internal Rules

Internal rules are defined on Cisco IOS XR Software and are used by protocols and processes within Cisco IOS XR Software. These rules are not customer configurable, but you may view them by using the **show logging correlator rule** command. All internal rule names are prefixed with [INTERNAL].

SNMP Alarm Correlation

In large-scale systems, such as Cisco IOS XR multi-chassis system , there may be situations when you encounter many SNMP traps emitted at regular intervals of time. These traps, in turn, cause additional time in the Cisco IOS XR processing of traps.

The additional traps can also slow down troubleshooting and increases workload for the monitoring systems and the operators. So, this feature addresses these issues.

The objective of this SNMP alarm correlation feature is to:

- Extract the generic pieces of correlation functionality from the existing syslog correlator
- Create DLLs and APIs suitable for reusing the functionality in other components
- Integrate the SNMP agent with the DLLs to enable SNMP trap correlation

How to Implement and Monitor Alarm Management and Logging Correlation

Configuring Logging Correlation Rules

This task explains how to configure logging correlation rules.

The purpose of configuring logging correlation rules is to define the root cause and non-root-cause alarm messages (with message category, group, and code combinations) for logging correlation. The originating root-cause alarm message is forwarded to the syslog process, and all subsequent (non-root-cause) alarm messages are sent to the logging correlation buffer.

The fields inside a message that can be used for configuring correlation rules are as follows:

- Message category (for example, PKT_INFRA, MGBL, OS)
- Message group (for example, LINK, LINEPROTO, or OIR)
- Message code (for example, UPDOWN or GO_ACTIVE).

The logging correlator mechanism, running on the active route processor, begins queueing messages matching the ones specified in the correlation rules for the time specified in the timeout interval of the correlation rule.

The timeout interval begins when the correlator captures any alarm message specified for a given rule.

SUMMARY STEPS

1. **configure**
2. **logging correlator rule** *correlation-rule* { **type** { **stateful** | **nonstateful** }}
3. **timeout** [*milliseconds*]
4. Use the **commit** or **end** command.
5. **show logging correlator rule** {**all** | *correlation-rule1 ... correlation-rule14* } [**context** *context1 ... context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator rule** *correlation-rule* { **type** { **stateful** | **nonstateful** }} **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator rule rule_stateful`	Configures a logging correlation rule. • Stateful correlations can change specifically if the root-cause alarm is bistate. • Nonstate correlations cannot change. All non-root-cause alarms remain in the correlation buffers.
Step 3	**timeout** [*milliseconds*] **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st)# timeout 60000`	Specifies the collection period duration time for the logging correlator rule message. • Timeout begins when the first alarm message identified by the correlation rule is logged.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	**show logging correlator rule** {**all** | *correlation-rule1 ... correlation-rule14* } [**context** *context1 ... context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**] **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule all`	(Optional) Displays defined correlation rules. • The output describes the configuration of each rule name, including the message category, group, and code information.

Configuring Logging Correlation Rule Sets

This task explains how to configure logging correlation rule sets.

SUMMARY STEPS

1. **configure**
2. **logging correlator ruleset** *ruleset*
3. **rulename** *rulename*
4. Use the **commit** or **end** command.
5. **show logging correlator ruleset** { **all** | *correlation-ruleset1...correlation-ruleset14* } [**detail** | **summary**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator ruleset** *ruleset* **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator ruleset ruleset1`	Configures a logging correlation rule set.
Step 3	**rulename** *rulename* **Example:** `RP/0/RSP0/CPU0:router(config-corr-ruleset)# rulename stateful_rule`	Configures a rule name.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	**show logging correlator ruleset** { **all**	*correlation-ruleset1...correlation-ruleset14* } [**detail**	**summary**] **Example:**	(Optional) Displays defined correlation rule sets.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# show logging correlator ruleset all`	

Configuring Root-cause and Non-root-cause Alarms

To correlate a root cause to one or more non-root-cause alarms and configure them to a rule, use the **rootcause** and **nonrootcause** commands specified for the correlation rule.

SUMMARY STEPS

1. **configure**
2. **logging correlator rule** *correlation-rule* { **type** { **stateful** | **nonstateful** }}
3. **rootcause** { *msg-category group-name msg-code* }
4. **nonrootcause**
5. **alarm** *msg-category group-name msg-code*
6. Use the **commit** or **end** command.
7. **show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator rule** *correlation-rule* { **type** { **stateful**	**nonstateful** }} **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator rule rule_stateful`	Configures a logging correlation rule and enters submodes for stateful and nonstateful rule types. • Stateful correlations can change specifically if the root-cause alarm is bistate. • Nonstate correlations cannot change. All non-root-cause alarms remain in the correlation buffers.
Step 3	**rootcause** { *msg-category group-name msg-code* } **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st)# rootcause CAT_BI_1 GROUP_BI_1 CODE_BI_1`	Configures a root-cause alarm message. • This example specifies a root-cause alarm under stateful configuration mode

	Command or Action	Purpose
Step 4	**nonrootcause** **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st)# nonrootcause`	Enters the non-root-cause configuration mode
Step 5	**alarm** *msg-category group-name msg-code* **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st-nonrc)# alarm CAT_BI_2 GROUP_BI_2 CODE_BI_2`	Specifies a non-root-cause alarm message. • This command can be issued with the **nonrootcause** command, such as **nonrootcause alarm** *msg-category group-name msg-code*
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 7	**show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**] **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule all`	(Optional) Displays the correlator rules that are defined.

Configuring Hierarchical Correlation Rule Flags

Hierarchical correlation is when a single alarm is both a root cause for one correlation rule and a non-root cause for another rule, and when alarms are generated resulting in a successful correlation associated with both rules. What happens to a non-root-cause alarm hinges on the behavior of its correlated root-cause alarm.

There are cases in which you want to control the stateful behavior associated with these hierarchies and to implement flags, such as reparenting and reissuing of nonbistate alarms. This task explains how to implement these flags.

See the Reparent Flag, on page 12 and Reissue Nonbistate Flag, on page 12 sections for detailed information about these flags.

SUMMARY STEPS

1. **configure**
2. **logging correlator rule** *correlation-rule* { **type** { **stateful** | **nonstateful** }}
3. **reissue-nonbistate**
4. **reparent**
5. Use the **commit** or **end** command.
6. **show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator rule** *correlation-rule* { **type** { **stateful** | **nonstateful** }} **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator rule rule_stateful type nonstateful`	Configures a logging correlation rule. • Stateful correlations can change specifically if the root-cause alarm is bistate. • Nonstateful correlations cannot change. All non-root-cause alarms remain in the correlation buffers.
Step 3	**reissue-nonbistate** **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st)# reissue-nonbistate`	Issues nonbistate alarm messages (events) from the correlator log after its root-cause alarm clears.
Step 4	**reparent** **Example:** `RP/0/RSP0/CPU0:router(config-corr-rule-st)# reparent`	Specifies the behavior of non-root-cause alarms after a root-cause parent clears.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	**show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**] **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule all`	(Optional) Displays the correlator rules that are defined.

What to do next

To activate a defined correlation rule and rule set, you must apply them by using the **logging correlator apply rule** and **logging correlator apply ruleset** commands.

Applying Logging Correlation Rules

This task explains how to apply logging correlation rules.

Applying a correlation rule activates it and gives a scope. A single correlation rule can be applied to multiple scopes on the router; that is, a rule can be applied to the entire router, to several locations, or to several contexts.

Note When a rule is applied or if a rule set that contains this rule is applied, then the rule definition cannot be modified through the configuration until the rule or rule set is once again unapplied.

Note It is possible to configure apply settings at the same time for both a rule and rule sets that contain the rule. In this case, the apply settings for the rule are the union of all these apply configurations.

SUMMARY STEPS

1. **configure**
2. **logging correlator apply rule** *correlation-rule*
3. Do one of the following:
   - **all-of-router**
   - **location** *node-id*
   - **context** *name*
4. Use the **commit** or **end** command.

5. **show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator apply rule** *correlation-rule* **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator apply-rule rule1`	Applies and activates a correlation rule and enters correlation apply rule configuration mode.

Step 3	Do one of the following:
	• **all-of-router**
	• **location** *node-id*
	• **context** *name*
	Example:
	`RP/0/RSP0/CPU0:router(config-corr-apply-rule)# all-of-router`
	or
	`RP/0/RSP0/CPU0:router(config-corr-apply-rule)# location 0/2/CPU0`
	or
	`RP/0/RSP0/CPU0:router(config-corr-apply-rule)# logging correlator apply-rule rule2 context POS_0_0_0_0`
		• Applies a logging correlation rule to all nodes on the router.
		• Applies a logging correlation rule to a specific node on the router.
		  • The location of the node is specified in the format *rack/slot/module*.
		• Applies a logging correlation rule to a specific context.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	**show logging correlator rule** { **all** | *correlation-rule1...correlation-rule14* } [**context** *context1...context 6*] [**location** *node-id1...node-id6*] [**rulesource** { **internal** | **user** }] [**ruletype** { **nonstateful** | **stateful** }] [**summary** | **detail**] **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule all`	(Optional) Displays the correlator rules that are defined.

Applying Logging Correlation Rule Sets

This task explains how to apply logging correlation rule sets.

Applying a correlation rule set activates it and gives a scope. When applied, a single rule-set configuration immediately effects the rules that are part of that given rule set.

Note Rule definitions that were previously applied (singly or as part of another rule set) cannot be modified until that rule or rule set is unapplied. Use the **no** form of the command to negate usage and then try to reapply rule set.

SUMMARY STEPS

1. **configure**
2. **logging correlator apply ruleset** *correlation-rule*
3. Do one of the following:
   - **all-of-router**
   - **location** *node-id*
   - **context** *name*
4. Use the **commit** or **end** command.
5. **show logging correlator ruleset** { **all** | *correlation-ruleset1 ... correlation-ruleset14* } [**detail** | **summary**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# configure`	
Step 2	**logging correlator apply ruleset** *correlation-rule* **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator apply ruleset ruleset2`	Applies and activates a rule set and enters correlation apply rule set configuration mode.

Step 3	Do one of the following:
	• **all-of-router**
	• **location** *node-id*
	• **context** *name*
	Example:
	`RP/0/RSP0/CPU0:router(config-corr-ruleset)# all-of-router`
	or
	`RP/0/RSP0/CPU0:router(config-corr-ruleset)# location 0/2/CPU0`
	or
	`RP/0/RSP0/CPU0:router(config-corr-ruleset)# context`
		• Applies a logging correlation rule set to all nodes on the router.
		• Applies a logging correlation rule set to a specific node on the router.
		  • The location of the node is specified in the format *rack/slot/module* .
		• Applies a logging correlation rule set to a specific context.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	**show logging correlator ruleset** { **all** | *correlation-ruleset1 ... correlation-ruleset14* } [**detail** | **summary**] **Example:** `RP/0/RSP0/CPU0:router# show logging correlator ruleset all`	(Optional) Displays the correlator rules that are defined.

Modifying Logging Events Buffer Settings

Logging events buffer settings can be adjusted to respond to changes in user activity, network events, or system configuration events that affect network performance, or in network monitoring requirements. The appropriate settings depend on the configuration and requirements of the system.

This task involves the following steps:

- Modifying logging events buffer size
- Setting threshold for generating alarms
- Setting the alarm filter (severity)

Caution Modifications to alarm settings that lower the severity level for reporting alarms and threshold for generating capacity-warning alarms may slow system performance.

Caution Modifying the logging events buffer size clears the buffer of all event records except for the bistate alarms in the set state.

SUMMARY STEPS

1. **show logging events info**
2. **configure**
3. **logging events buffer-size** *bytes*
4. **logging events threshold** *percent*
5. **logging events level** *severity*
6. Use the **commit** or **end** command.
7. **show logging events info**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events info** **Example:** `RP/0/RSP0/CPU0:router# show logging events info`	(Optional) Displays the size of the logging events buffer (in bytes), the percentage of the buffer that is occupied by alarm-event records, capacity threshold for reporting alarms, total number of records in the buffer, and severity filter, if any.
Step 2	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**logging events buffer-size** *bytes* **Example:** `RP/0/RSP0/CPU0:router(config)# logging events buffer-size 50000`	Specifies the size of the alarm record buffer. • In this example, the buffer size is set to 50000 bytes.
Step 4	**logging events threshold** *percent* **Example:** `RP/0/RSP0/CPU0:router(config)# logging events threshold 85`	Specifies the percentage of the logging events buffer that must be filled before the alarm logger generates a threshold-crossing alarm. • In this example, the alarm logger generates athreshold-crossing alarm notification when the event buffer reaches 85 percent of capacity.
Step 5	**logging events level** *severity* **Example:** `RP/0/RSP0/CPU0:router(config)# logging events level warnings`	Sets the severity level that determines which logging events are displayed. (See Table 2: Alarm Severity Levels for Event Logging , on page 10 under the Alarm Severity Level and Filtering, on page 10 section for a list of the severity levels.) • Keyword options are as follows: **emergencies**, **alerts**, **critical**, **errors**, **warnings**, **notifications**, and **informational**. • In this example, messages with a warning (Level 4) severity or greater are written to the alarm log. Messages of a lesser severity (notifications and informational messages) are not recorded.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 7	**show logging events info** **Example:** `RP/0/RSP0/CPU0:router# show logging events info`	(Optional) Displays the size of the logging events buffer (in bytes), percentage of the buffer that is occupied by alarm-event records, capacity threshold for reporting alarms, total number of records in the buffer, and severity filter, if any. • This command is used to verify that all settings have been modified and that the changes have been accepted by the system.

Modifying Logging Correlator Buffer Settings

This task explains how to modify the logging correlator buffer settings.

The size of the logging correlator buffer can be adjusted to accommodate the anticipated volume of incoming correlated messages. Records can be removed from the buffer by correlation ID, or the buffer can be cleared of all records.

SUMMARY STEPS

1. **configure**
2. **logging correlator buffer-size** *bytes*
3. **exit**
4. **show logging correlator info**
5. **clear logging correlator delete** *correlation-id*
6. **clear logging correlator delete all-in-buffer**
7. **show logging correlator buffer** { **all-in-buffer** [**ruletype** [**nonstateful** | **stateful**]] | [**rulesource** [**internal** | **user**]] | **rule-name** *correlation-rule1...correlation-rule14* | **correlationID** *correlation-id1..correlation-id14* }

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**logging correlator buffer-size** *bytes* **Example:** `RP/0/RSP0/CPU0:router(config)# logging correlator buffer-size 100000`	Specifies the size of the logging correlator buffer. • In this example, the size of the logging correlator buffer is set to 100,000 bytes.
Step 3	**exit** **Example:** `RP/0/RSP0/CPU0:router(config)# exit`	Exits global configuration mode and returns the router to EXEC mode.
Step 4	**show logging correlator info** **Example:** `RP/0/RSP0/CPU0:router# show logging correlator info`	(Optional) Displays information about the size of the logging correlator buffer and percentage of the buffer occupied by correlated messages

	Command or Action	Purpose
Step 5	**clear logging correlator delete** *correlation-id* **Example:** `RP/0/RSP0/CPU0:router# clear logging correlator delete 48 49 50`	(Optional) Removes a particular correlated event record or records from the logging correlator buffer. • A range of correlation IDs can also be specified for removal (up to 32 correlation IDs, separated by a space).
Step 6	**clear logging correlator delete all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# clear logging correlator delete all-in-buffer`	(Optional) Clears all correlated event messages from the logging correlator buffer.
Step 7	**show logging correlator buffer** { **all-in-buffer** [**ruletype** [**nonstateful** | **stateful**]] | [**rulesource** [**internal** | **user**]] | **rule-name** *correlation-rule1...correlation-rule14* | **correlationID** *correlation-id1..correlation-id14* } **Example:** `RP/0/RSP0/CPU0:router# show logging correlator buffer all-in-buffer`	(Optional) Displays the contents of the correlated event record. • Use this step to verify that records for particular correlation IDs have been removed from the correlated event log.

Displaying Alarms by Severity and Severity Range

This task explains how to display alarms by severity and severity range.

Alarms can be displayed according to severity level or a range of severity levels. Severity levels and their respective system conditions are listed in Table 2: Alarm Severity Levels for Event Logging , on page 10 under the Alarm Severity Level and Filtering, on page 10 section.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer severity-lo-limit** *severity*
2. **show logging events buffer severity-hi-limit** *severity*
3. **show logging events buffer severity-hi-limit** *severity* **severity-lo-limit** *severity*
4. **show logging events buffer severity-hi-limit** *severity* **severity-lo-limit** *severity* **timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer severity-lo-limit** *severity* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer severity-lo-limit notifications`	(Optional) Displays logging events with a severity at or below the numeric value of the specified severity level. • In this example, alarms with a severity of notifications (severity of 5) or lower are displayed. Informational (severity of 6) messages are omitted. **Note** Use the **severity-lo-limit** keyword and the *severity* argument to specify the severity level *description*, not the numeric value assigned to that severity level.
Step 2	**show logging events buffer severity-hi-limit** *severity* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer severity-hi-limit critical`	(Optional) Displays logging events with a severity at or above the numeric value specified severity level. • In this example, alarms with a severity of critical (severity of 2) or greater are displayed. Alerts (severity of 1) and emergencies (severity of 0) are omitted. **Note** Use the **severity-hi-limit** keyword and the *severity* argument to specify the severity level *description*, not the numeric value assigned to that severity level.
Step 3	**show logging events buffer severity-hi-limit** *severity* **severity-lo-limit** *severity* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer severity-hi-limit alerts severity-lo-limit critical`	(Optional) Displays logging events within a severity range. • In this example, alarms with a severity of critical (severity of 2) and alerts (severity of 1) are displayed. All other event severities are omitted.
Step 4	**show logging events buffer severity-hi-limit** *severity* **severity-lo-limit** *severity* **timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer severity-lo-limit warnings severity-hi-limit critical timestamp-lo-limit 22:00:00 may 07 04`	(Optional) Displays logging events occurring after the specified time stamp and within a severity range. The *month*, *day*, and *year* arguments default to the current month, date, and year, if not specified. • In this example, alarms with a severity of warnings (severity of 4), errors (severity of 3), and critical (severity of 2) that occur after 22:00:00 on May 7, 2004 are displayed. All other messages occurring before the time stamp are omitted.

Displaying Alarms According to a Time Stamp Range

Alarms can be displayed according to a time stamp range. Specifying a specific beginning and endpoint can be useful in isolating alarms occurring during a particular known system event.

This task explains how to display alarms according to a time stamp range.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*]
2. **show logging events buffer timestamp-hi-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*]
3. **show logging events buffer timestamp-hi-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer timestamp-lo-limit 21:28:00 april 18 04`	(Optional) Displays logging events with a time stamp after the specified time and date. • The *month*, *day*, and *year* arguments default to the current month, date, and year if not specified. • The sample output displays events logged after 21:28:00 on April 18, 2004.
Step 2	**show logging events buffer timestamp-hi-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer timestamp-hi-limit 21:28:03 april 18 04`	(Optional) Displays logging events with a time stamp before the specified time and date. • The *month*, *day*, and *year* arguments default to the current month, date, and year if not specified. • The sample output displays events logged before 21:28:03 on April 18, 2004.
Step 3	**show logging events buffer timestamp-hi-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer timestamp-hi-limit 21:28:00 april 18 04 timestamp-lo-limit 21:16:00 april 18 03`	(Optional) Displays logging events with a time stamp after and before the specified time and date. • The *month*, *day*, and *year* arguments default to the current month, day, and year if not specified. • The sample output displays events logged after 21:16:00 on April 18, 2003 and before 21:28:00 on April 18, 2004.

Displaying Alarms According to Message Group and Message Code

This task explains how to display alarms in the logging events buffer according to message code and message group.

Displaying alarms by message group and message code can be useful in isolating related events.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer group** *message-group*
2. **show logging events buffer message** *message-code*
3. **show logging events buffer group** *message-group* **message** *message-code*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer group** *message-group* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer group SONET`	(Optional) Displays logging events matching the specified message group. • In this example, all events that contain the message group SONET are displayed.
Step 2	**show logging events buffer message** *message-code* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer message ALARM`	(Optional) Displays logging events matching the specified message code. • In this example, all events that contain the message code ALARM are displayed.
Step 3	**show logging events buffer group** *message-group* **message** *message-code* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer group SONET message ALARM`	(Optional) Displays logging events matching the specified message group and message code. • In this example, all events that contain the message group SONET and message code ALARM are displayed.

Displaying Alarms According to a First and Last Range

This task explains how to display alarms according to a range of the first and last alarms in the logging events buffer.

Alarms can be displayed according to a range, beginning with the first or last alarm in the logging events buffer.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer first** *event-count*
2. **show logging events buffer last** *event-count*
3. **show logging events buffer first** *event-count* **last** *event-count*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer first** *event-count* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer first 15`	(Optional) Displays logging events beginning with the first event in the logging events buffer. • For the *event-count* argument, enter the number of events to be displayed. • In this example, the first 15 events in the logging events buffer are displayed.
Step 2	**show logging events buffer last** *event-count* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer last 20`	(Optional) Displays logging events beginning with the last event in the logging events buffer. • For the *event-count* argument, enter the number of events to be displayed. • In this example, the last 20 events in the logging events buffer are displayed.
Step 3	**show logging events buffer first** *event-count* **last** *event-count* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer first 20 last 20`	(Optional) Displays the first and last events in the logging events buffer. • For the *event-count* argument, enter the number of events to be displayed. • In this example, both the first 20 and last 20 events in the logging events buffer are displayed.

Displaying Alarms by Location

This task explains how to display alarms by location.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer location** *node-id*
2. **show logging events buffer location** *node-id* **event-hi-limit** *event-id* **event-lo-limit** *event-id*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer 0/2/CPU0`	(Optional) Isolates the occurrence of the range of event IDs to a particular node. • The location of the node is specified in the format *rack/slot/module*.
Step 2	**show logging events buffer location** *node-id* **event-hi-limit** *event-id* **event-lo-limit** *event-id* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer location 0/2/CPU0 event-hi-limit 100 event-lo-limit 1`	(Optional) Isolates the occurrence of the range of event IDs to a particular node and narrows the range by specifying a high and low limit of event IDs to be displayed. • The location of the node is specified in the format *rack/slot/module*.

Displaying Alarms by Event Record ID

This task explains how to display alarms by event record ID.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer all-in-buffer**
2. **show logging events buffer event-hi-limit** *event-id* **event-lo-limit** *event-id*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer all-in-buffer`	(Optional) Displays all messages in the logging events buffer. **Caution** Depending on the alarm severity settings, use of this command can create a large amount of output.

	Command or Action	Purpose
Step 2	**show logging events buffer event-hi-limit** *event-id* **event-lo-limit** *event-id* **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer event-hi-limit 100 event-lo-limit 1`	(Optional) Narrows the range by specifying a high and low limit of event IDs to be displayed.

Displaying the Logging Correlation Buffer Size, Messages, and Rules

This task explains how to display the logging correlation buffer size, messages in the logging correlation buffer, and correlation rules.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging correlator info**
2. **show logging correlator buffer all-in-buffer**
3. **show logging correlator buffer correlationID** *correlation-id*
4. **show logging correlator buffer rule-name** *correlation-rule*
5. **show logging correlator rule all**
6. **show logging correlator rule** *correlation-rule*
7. **show logging correlator ruleset all**
8. **show logging correlator ruleset** *ruleset-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging correlator info** **Example:** `RP/0/RSP0/CPU0:router# show logging correlator info`	(Optional) Displays the size of the logging correlation buffer (in bytes) and the percentage occupied by correlated messages.
Step 2	**show logging correlator buffer all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# show logging correlator buffer all-in-buffer`	(Optional) Displays all messages in the logging correlation buffer.

	Command or Action	Purpose
Step 3	**show logging correlator buffer correlationID** *correlation-id* **Example:** `RP/0/RSP0/CPU0:router# show logging correlator buffer correlationID 37`	(Optional) Displays specific messages matching a particular correlation ID in the correlation buffer.
Step 4	**show logging correlator buffer rule-name** *correlation-rule* **Example:** `RP/0/RSP0/CPU0:router# show logging correlator buffer rule-name rule7`	(Optional) Displays specific messages matching a particular rule in the correlation buffer.
Step 5	**show logging correlator rule all** **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule all`	(Optional) Displays all defined correlation rules.
Step 6	**show logging correlator rule** *correlation-rule* **Example:** `RP/0/RSP0/CPU0:router# show logging correlator rule rule7`	(Optional) Displays the specified correlation rule.
Step 7	**show logging correlator ruleset all** **Example:** `RP/0/RSP0/CPU0:router# show logging correlator ruleset all`	(Optional) Displays all defined correlation rule sets.
Step 8	**show logging correlator ruleset** *ruleset-name* **Example:** `RP/0/RSP0/CPU0:router# show logging correlator ruleset ruleset_static`	(Optional) Displays the specified correlation rule set.

Clearing Alarm Event Records and Resetting Bistate Alarms

This task explains how to clear alarm event records and bistate alarms.

Unnecessary and obsolete messages can be cleared to reduce the size of the event logging buffer and make it more searchable, and thus more navigable.

The filtering capabilities available for clearing events in the logging events buffer (with the **clear logging events delete** command) are also available for displaying events in the logging events buffer (with the **show logging events buffer** command).

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show logging events buffer all-in-buffer**
2. **clear logging events delete timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*]
3. **clear logging events delete event-hi-limit** *severity* **event-lo-limit** *severity*
4. **clear logging events delete location** *node-id*
5. **clear logging events delete first** *event-count*
6. **clear logging events delete last** *event-count*
7. **clear logging events delete message** *message-code*
8. **clear logging events delete group** *message-group*
9. **clear logging events reset all-in-buffer**
10. **show logging events buffer all-in-buffer**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show logging events buffer all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer all-in-buffer`	It retains the messages before the specified time and displayed the messages after the timestamp. The timestamp-lo-limit specifies the lower time limit. Similarly timestamp-hi-limit specifies the higher time limit of a time window. All events within this time window will be displayed. The default value of the timestamp-lo-limit is the timestamp of the earliest event in the buffer. The timestamp-hi-limit is the timestamp of the latest event in the buffer.
Step 2	**clear logging events delete timestamp-lo-limit** *hh* **:** *mm* **:** *ss* [*month*] [*day*] [*year*] **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete timestamp-lo-limit 20:00:00 april 01 2004`	It retains the messages before the specified time and deletes the messages after the timestamp. The timestamp-lo-limit specifies the lower time limit. Similarly timestamp-hi-limit specifies the higher time limit of a time window. All events within this time window will be deleted. The default value of the timestamp-lo-limit is the timestamp of the earliest event in the buffer. The timestamp-hi-limit is the timestamp of the latest event in the buffer.
Step 3	**clear logging events delete event-hi-limit** *severity* **event-lo-limit** *severity* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete event-hi-limit warnings event-lo-limit informational`	(Optional) Deletes logging events within a range of severity levels for logging alarm messages. • In this example, all events with a severity level of warnings, notifications, and informational are deleted.

	Command or Action	Purpose
Step 4	**clear logging events delete location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete location 0/2/CPU0`	(Optional) Deletes logging events from the logging events that have occurred on a particular node. • The location of the node is specified in the format *rack/slot/module*.
Step 5	**clear logging events delete first** *event-count* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete first 10`	(Optional) Deletes logging events beginning with the first event in the logging events buffer. • In this example, the first 10 events in the logging events buffer are cleared.
Step 6	**clear logging events delete last** *event-count* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete last 20`	(Optional) Deletes logging events beginning with the last event in the logging events buffer. • In this example, the last 20 events in the logging events buffer are cleared.
Step 7	**clear logging events delete message** *message-code* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete message sys`	(Optional) Deletes logging events that contain the specified message code. • In this example, all events that contain the message code SYS are deleted from the logging events buffer.
Step 8	**clear logging events delete group** *message-group* **Example:** `RP/0/RSP0/CPU0:router# clear logging events delete group config_i`	(Optional) Deletes logging events that contain the specified message group. • In this example, all events that contain the message group CONFIG_I are deleted from the logging events buffer.
Step 9	**clear logging events reset all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# clear logging events reset all-in-buffer`	(Optional) Clears all bistate alarms in the SET state from the logging events buffer.
Step 10	**show logging events buffer all-in-buffer** **Example:** `RP/0/RSP0/CPU0:router# show logging events buffer all-in-buffer`	(Optional) Displays all messages in the logging events buffer.

Defining SNMP Correlation Buffer Size

This task explains how to define correlation buffer size for SNMP traps.

SUMMARY STEPS

1. **configure**
2. **snmp-server correlator buffer-size** *bytes*
3. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server correlator buffer-size** *bytes* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server correlator buffer-size 600`	Defines the buffer size that can store SNMP correlation traps. The default size is 64KB. You can clear the correlation buffers manually or the buffer wraps automatically, wherein the oldest correlations are purged to accommodate the newer correlations.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Defining SNMP Rulesets

This task defines a ruleset that allows you to group two or more rules into a group. You can apply the specified group to a set of hosts or all of them.

SUMMARY STEPS

1. **configure**
2. **snmp-server correlator ruleset** *name* **rulename** *name*
3. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server correlator ruleset** *name* **rulename** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server correlator ruleset rule1 rulename rule2 host ipv4 address 1.2.3.4 host ipv4 address 2.3.4.5 port 182`	Specifies a ruleset that allows you to group two or more rules into a group and apply that group to a set of hosts.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring SNMP Correlation Rules

This task explains how to configure SNMP correlation rules.

The purpose of configuring SNMP trap correlation rules is to define the correlation rules or non-correlation rules and apply them to specific trap destinations.

SUMMARY STEPS

1. **configure**
2. **snmp-server correlator rule** *rule_name* { **nonrootcause trap** *trap_oid* **varbind** *vbind_OID* { **index** | **value** } **regex** *line* | **rootcause trap** *trap_oid* **varbind** *vbind_OID* { **index** | **value** } **regex** *line* | **timeout** }
3. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server correlator rule** *rule_name* { **nonrootcause trap** *trap_oid* **varbind** *vbind_OID* { **index** | **value** } **regex** *line* | **rootcause trap** *trap_oid* **varbind** *vbind_OID* { **index | value** } **regex** *line* | **timeout** } **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server correlator rule test rootcause A varbind A1 value regex RA1 varbind A2 index regex RA2 timeout 5000 nonrootcause trap B varbind B1 index regex RB1 varbind B2 value regex RB2 trap C varbind C1 value regex RC1 varbind C2 value regex RC2`	Configures a SNMP correlation rule. You can specify the numeric rootcause trap OID or non-rootcause trap matching definitions. • Specifies a numeric non-rootcause trap OID and, optionally, one or more numeric varbinds specific to the non-rootcause trap that must ALL also be matched to have found a valid non-rootcause for this rule. The POSIX regexp specifies a regular expression that the value that the vbind index or value must match. • Specifies a numeric rootcause trap OID and, optionally, one or more numeric varbinds specific to the rootcause trap that must ALL also be matched to have found a valid rootcause for this rule. The POSIX regexp specifies a regular expression that the vbind index or value must match. **Note** You can specify the timeout for detection of a correlation after receipt of first rootcause or non-rootcause in this specified rule. The range is from 1 to 600000 milliseconds. **Note** All OID values for traps and varbinds are verified and rejected, if they do not match valid OIDs supported by IOS XR.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Applying SNMP Correlation Rules

The purpose of this task is to apply the SNMP trap correlation rules to specific trap destinations.

SUMMARY STEPS

1. **configure**
2. **snmp-server correlator apply rule** *rule-name* [**all-hosts** | **host ipv4 address** *address* [*port*]
3. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server correlator apply rule** *rule-name* [**all-hosts** | **host ipv4 address** *address* [*port*] **Example:** `RP/0/RSP0/CPU0:router# snmp-server correlator apply rule ifupdown host ipv4 address 1.2.3.4 host ipv4 address 2.3.4.5 port 182`	Applies the SNMP trap correlation rules to specific trap destinations. You have an option of applying the rule to traps destined for all trap hosts, or to a specific subset by specifying individual IP addresses and optional ports.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Applying SNMP Correlation Ruleset

The purpose of this task is to apply the set of two SNMP trap correlation rules or more rules as a group to specific trap destinations.

SUMMARY STEPS

1. **configure**
2. **snmp-server correlator apply ruleset** *ruleset-name* [**all-hosts** | **host ipv4 address** *address* [*port*]
3. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server correlator apply ruleset** *ruleset-name* [**all-hosts** | **host ipv4 address** *address* [*port*] **Example:** `RP/0/RSP0/CPU0:router# snmp-server correlator apply ruleset ruleset_1 host ipv4 address 1.2.3.4 host ipv4 address 2.3.4.5 port 182`	Applies the SNMP trap correlation ruleset to specific trap destinations. You have an option of applying the set of two or more SNMP trap correlation rules to traps destined for all trap hosts, or to a specific subset by specifying individual IP addresses and optional ports.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Asynchronous Syslog Communication

The asynchronous syslog communication feature enables proper ordering of messages testing on each node (LC, RP), non dropping of messages generated from multiple clients on each node (LC, RP) and checking performance, scalability and latency by sending log messages at incremental rates.

This feature enables the following:

- Proper ordering of messages testing on MC min 4+1.
- Non dropping of messages generated from multiple clients on MC min 4+1.

- Syslogd_helper message handling capacity - flood lots of syslog messages using test client (logger), verify if no syslog message is lost (specified rate as per new design).
- 1200/1500 msgs/sec from every node - restart restart/crash syslogd_helper on LCs and RP/correlatord and syslogd on RP.
- Configure the routing protocol ospf. Configure 5k neighbors using sub interfaces. Perform interface flapping to generate log messages and check syslogd_helper performance.
- Enable debug for few heavy processes - sysdb/gsp

Configuration Examples for Alarm Management and Logging Correlation

This section provides these configuration examples:

Increasing the Severity Level for Alarm Filtering to Display Fewer Events and Modifying the Alarm Buffer Size and Capacity Threshold: Example

This configuration example shows how to set the capacity threshold to 90 percent, to reduce the size of the logging events buffer to 10,000 bytes from the default, and to increase the severity level to errors:

```
!
logging events threshold 90
logging events buffer-size 10000
logging events level errors
!
```

Increasing the severity level to errors reduces the number of alarms that are displayed in the logging events buffer, because only alarms with a severity of errors or higher are displayed. Increasing the threshold capacity to 90 percent reduces the time interval between the threshold crossing and wraparound events; the logging events buffer thus does not generate a threshold-crossing alarm until it reaches 90 percent capacity. Reducing the size of the logging events buffer to 10,000 bytes decreases the number of alarms that are displayed in the logging events buffer and reduces the memory requirements for the component.

Configuring a Nonstateful Correlation Rule to Permanently Suppress Node Status Messages: Example

This example shows how to configure a nonstateful correlation rule to permanently suppress node status messages:

```
logging correlator rule node_status type nonstateful
timeout 4000
  rootcause PLATFORM INVMGR NODE_STATE_CHANGE
  nonrootcause
    alarm PLATFORM SYSLDR LC_ENABLED
    alarm PLATFORM ALPHA_DISPLAY CHANGE
  !
!
```

```
logging correlator apply rule node_status

  all-of-router
!
```

In this example, three similar messages are identified as forwarded to the syslog process simultaneously after a card boots:

PLATFORM-INVMGR-6-NODE_STATE_CHANGE : Node: 0/1/CPU0, state: IOS XR RUN

PLATFORM-SYSLDR-5-LC_ENABLED : LC in slot 1 is now running IOX

PLATFORM-ALPHA_DISPLAY-6-CHANGE : Alpha display on node 0/1/CPU0 changed to IOX RUN in state default

These messages are similar. To see only one message appear in the logs, one of the messages is designated as the root cause message (the one that appears in the logs), and the other messages are considered non-root-cause messages.

The root-cause message is typically the one that arrives earliest, but that is not a requirement.

```
logging correlator rule node_status type nonstateful
  timeout 4000
  rootcause PLATFORM INVMGR NODE_STATE_CHANGE
  nonrootcause
    alarm PLATFORM SYSLDR LC_ENABLED
    alarm PLATFORM ALPHA_DISPLAY CHANGE
  !
!
```

In this example, the correlation rule named node_status is configured to correlate the PLATFORM INVMGR NODE_STATE_CHANGE alarm (the root-cause message) with the PLATFORM SYSLDR LC_ENABLED and PLATFORM ALPHA_DISPLAY CHANGE alarms. The updown correlation rule is applied to the entire router.

```
logging correlator apply rule node_status
  all-of-router
!
```

After a card boots and sends these messages:

PLATFORM-INVMGR-6-NODE_STATE_CHANGE : Node: 0/1/CPU0, state: IOS XR RUN

PLATFORM-SYSLDR-5-LC_ENABLED : LC in slot 1 is now running IOX

PLATFORM-ALPHA_DISPLAY-6-CHANGE : Alpha display on node 0/1/CPU0 changed to IOX RUN in state default

the correlator forwards the PLATFORM-INVMGR-6-NODE_STATE_CHANGE message to the syslog process, while the remaining two messages are held in the logging correlator buffer.

In this example, the show sample output from the **show logging events buffer all-in-buffer** command displays the alarms stored in the logging events buffer after the 4-second time period expires for the node_status correlation rule:

```
RP/0/RSP0/CPU0:router# show logging events buffer all-in-buffer

#ID :C_id:Source :Time :%CATEGORY-GROUP-SEVERITY-MESSAGECODE: Text

#76 :12  :RP/0/0/CPU0:Aug  2 22:32:43 : invmgr[194]:
```

```
%PLATFORM-INVMGR-6-NODE_STATE_CHANGE : Node: 0/1/CPU0, state: IOS XR RUN
```

The **show logging correlator buffer** correlation ID command generates the following output after the one minute interval expires. The output displays the alarms assigned correlation ID 12 in the logging correlator buffer.

```
RP/0/RSP0/CPU0:router# show logging correlator buffer correlationID 46

#C_id.id:Rule Name:Source :Time : Text

#12.1 :nodestatus:RP/0/0/CPU0:Aug  2 22:32:43 : invmgr[194]:
%PLATFORM-INVMGR-6-NODE_STATE_CHANGE : Node: 0/1/CPU0, state: IOS XR RUN
#12.2 :nodestatus:RP/0/0/CPU0:Aug  2 22:32:43 : sysldr[336]: %PLATFORM-SYSLDR-5-LC_ENABLED
 : LC in slot 1 is now running IOX
#12.3 :nodestatus:RP/0/0/CPU0:Aug  2 22:32:44 : alphadisplay[102]:
%PLATFORM-ALPHA_DISPLAY-6-CHANGE : Alpha display on node 0/1/CPU0 changed to IOX RUN  in
state default
Because this rule was defined as nonstateful, these messages are held in the buffer
indefinitely.
```

Configuring a Stateful Correlation Rule for LINK UPDOWN and SONET ALARM Alarms: Example

This example shows how to configure a correlation rule for the LINK UPDOWN and SONET ALARM messages:

```
!
logging correlator rule updown type stateful
  timeout 10000
  rootcause PKT_INFRA LINK UPDOWN
  nonrootcause
    alarm L2 SONET ALARM
  !
!
logging correlator apply rule updown
  all-of-router
!
```

In this example, suppose that two routers are connected. When the correlator receives a root-cause message, the correlator sends it directly to the syslog process. Subsequent PKT_INFRA-LINK- UPDOWN or L2-SONET-ALARM messages matching the rule are considered leaf messages and are stored in the logging correlator buffer. If, for any reason, a leaf message (such as the L2-SONET-ALARM alarm in this example) is received first, the correlator does not send it to the logging events buffer immediately; the correlator, instead, waits until the timeout interval expires. After the timeout, if the root message is never received, all messages in the logging correlator buffer received during the timeout interval are forwarded to the syslog process.

In this example, the correlation rule named updown is configured to correlate the PKT_INFRA-LINK-UPDOWN alarm (the root message) and L2-SONET-ALARM alarms (leaf messages associated with PKT_INFRA-LINK-UPDOWN alarms).

```
logging correlator rule updown type stateful
  timeout 10000
  rootcause PKT_INFRA LINK UPDOWN
  nonrootcause
    alarm L2 SONET ALARM
In this example, the updown correlation rule is applied to the entire router:
```

```
logging correlator apply rule updown
  all-of-router
```

This example shows sample output from the **show logging events buffer all-in-buffer** command. The output displays the alarms stored in the logging events buffer after the one minute time period expires for the updown correlation rule configured:

```
RP/0/RSP0/CPU0:router# show logging events buffer all-in-buffer

#ID :C_id:Source :Time :%CATEGORY-GROUP-SEVERITY-MESSAGECODE: Text

#144 :46 :LC/0/7/CPU0:Jan 30 16:35:39 2004:ifmgr[130]: %PKT_INFRA-LINK-3-UPDOWN :

Interface POS0/7/0/0, changed state to Down
```

Note Only the first LINK UPDOWN root message is forwarded to the syslog process during the timeout interval.

The following example shows output from the **show logging correlator buffer correlationID** command generated after the one-minute interval expires. The output displays the alarms assigned correlation ID 46 in the logging correlator buffer. In the example, the PKT_INFRA-LINK-UPDOWN root-cause message and L2-SONET-ALARM leaf messages generated during the timeout interval assigned correlation ID 46 are displayed:

```
RP/0/RSP0/CPU0:router# show logging correlator buffer correlationID 46

#C_id.id:Rule Name:Source :Time : Text

#46.1 :updown :LC/0/7/CPU0:Jan 30 16:35:39 2004:ifmgr[130]: %PKT_INFRA-LINK-3-UPDOWN :

Interface POS0/7/0/0, changed state to Down

#46.2 :updown :LC/0/7/CPU0:Jan 30 16:35:41 2004:DI_Partner[50]: %L2-SONET-4-ALARM :

SONET0_7_0_0: SLOS
```

Note The subsequent PKT_INFRA-LINK-UPDOWN and L2-SONET-ALARM leaf messages generated during the timeout interval remain in the logging correlator buffer because they are leaf messages.

This example shows output from the **show logging correlator buffer correlationID** command. The output displays the alarms assigned to correlation IDs 46 and 47, the correlation IDs associated with the PKT_INFRA-LINK-UPDOWN and L2-SONET-ALARM root-cause messages:

```
RP/0/RSP0/CPU0:router# show logging correlator buffer correlationID 46

NO records matching query found
```

Pulse Queue State Monitoring

A pulse is a special notification mechanism whereby a sender sends a message to a receiving process without waiting for acknowledgment. For each pulse, memory is allocated from a separate kernel virtual space which

is used to hold kernel objects. If processes are not handling these pulse messages, the pulse queue count will increase which can lead to shortage in kernel virtual space and further pulse creation can fail and even lead to crash.

The pulse queue state is calculated for every 100 millisec which is the default interval to monitor CPU usage and other memory usage. The pulse queue state will switch from Normal to Minor / Severe / Critical if the pulse queue count increases beyond the defined threshold levels and traces will be collected. Once the pulse queue state switches from Normal to Severe/Critical, the top 5 pulse queue users or processes will be identified and alarm messages will be generated for every 1 minute with process info like pid, process name, etc.

Default threshold levels for pulse queue per system(node) are as below:

- 50000 number of pulses - (Critical)
- 30000 number of pulses - (Severe)
- 15000 number of pulses - (Minor)
- Less than 15000 - (Normal)

Additional References

The following sections provide references related to implementing and monitoring alarm logs and logging correlation on the Cisco ASR 9000 Series Router.

Related Documents

Related Topic	Document Title
Alarm and logging correlation commands	*Alarm Management and Logging Correlation Commands* module in the *System Monitoring Command Reference for Cisco ASR 9000 Series Routers*
Logging services commands	*Logging Services Commands* module in the *System Monitoring Command Reference for Cisco ASR 9000 Series Routers*
Onboard Failure Logging (OBFL) configuration tasks	*Implementing Logging Services* module in the *System Monitoring Command Reference for Cisco ASR 9000 Series Routers*
Onboard Failure Logging (OBFL) commands	*Onboard Failure Logging Commands* module in the *System Monitoring Command Reference for Cisco ASR 9000 Series Routers*
Cisco IOS XR software XML API material	*Cisco IOS XR XML API Guide*
Cisco IOS XR software getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of 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/cisco/web/support/index.html*