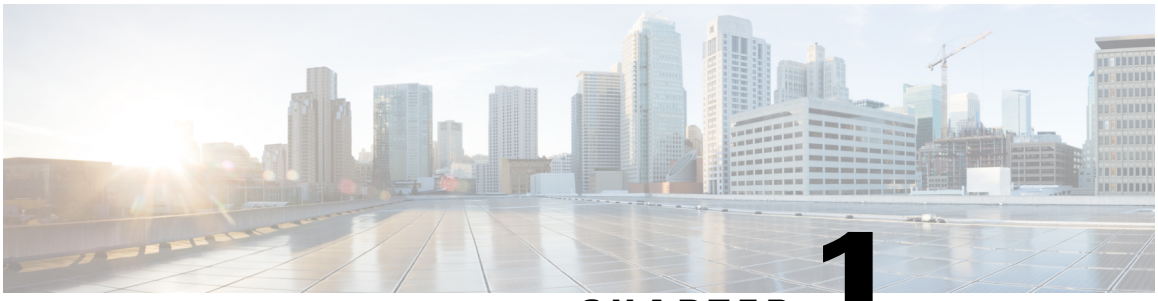
Embedded Resource Manager Configuration Guide, Cisco IOS XE Fuji 16.8.x

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

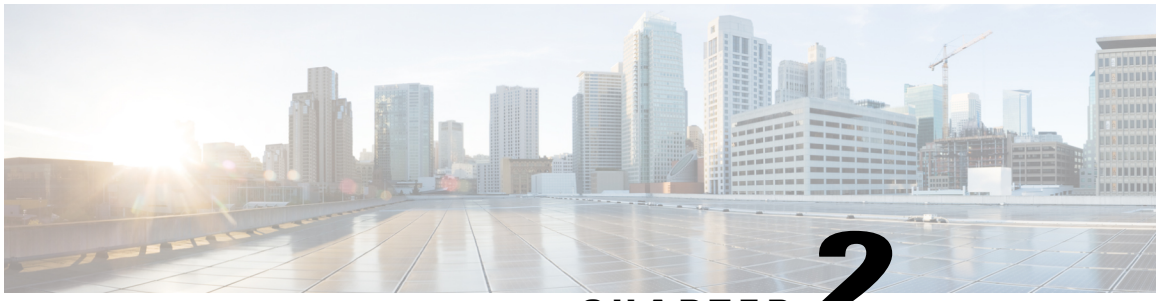
Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.



CHAPTER 2

Embedded Resource Manager (ERM)

The Embedded Resource Manager (ERM) feature allows you to monitor internal system resource utilization for specific resources such as the buffer, memory, and CPU. ERM monitors resource utilization from the perspective of various subsystems within the Cisco IOS software such as resource owners (ROs) and resource users (RUs). ERM allows you to configure threshold values for system resources.

The ERM infrastructure is designed to allow for granular monitoring on a task basis within the Cisco IOS software. Network administrators can define thresholds to create notifications according to the real-time resource consumption. ERM goes beyond simply monitoring for total CPU utilization. Through the use of ERM, network administrators and operators can gain a better understanding of the device's operational characteristics, leading to better insight into system scalability and improved system availability.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Embedded Resource Manager

You must be running Cisco IOS Release 12.4(6)T or a later release to use the Packet Memory Reclamation functionality.

Restrictions for Embedded Resource Manager

Additional instructions from a Cisco technical support representative may be required.

Information About Embedded Resource Manager

ERM promotes resource availability by providing the infrastructure to track resource usage.

To configure threshold values for resource manager entities, you should understand the following concepts:

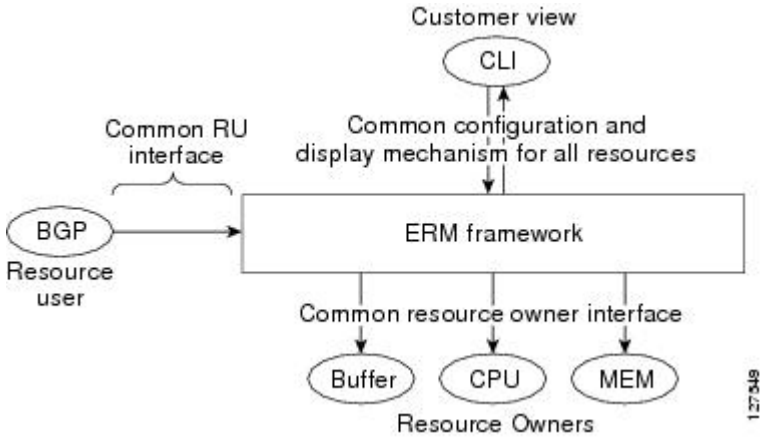
Benefits of the Embedded Resource Manager

The ERM framework tracks resource utilization and resource depletion by monitoring finite resources. Support for monitoring CPU, buffer, and memory utilization at a global or IOS-process level is available.

The ERM framework provides a mechanism to send notifications whenever the specified threshold values are exceeded by any resource user. This notification helps network administrators diagnose any CPU, buffer, and memory utilization issues.

The ERM architecture is illustrated in the figure below.

Figure 1: ERM Architecture



ERM provides a framework for monitoring any finite resource within the Cisco IOS software and provides information that a user can analyze to better understand how network changes might impact system operation. ERM helps in addressing infrastructure problems such as reloads, memory allocation failure, and high CPU utilization by performing the following functions:

- Monitoring system resource usage.
- Setting the resource threshold at a granular level.
- Generating alerts when resource utilization reaches the specified level.
- Generating internal events using the Cisco IOS Embedded Event Manager feature.

Resource Accounting and Thresholds Tracking in ERM

ERM tracks the resource usage for each RU internally. An RU is a subsystem or process task within the Cisco IOS software; for example, the Open Shortest Path First (OSPF) hello process is a resource user. Threshold limits are used to notify network operators of specific conditions. The ERM infrastructure provides a means to notify the internal RU subsystem of threshold indications as well. The resource accounting is performed by individual ROs. ROs are part of the Cisco IOS software and are responsible for monitoring certain resources such as the memory, CPU, and buffer. When the utilization for each RU exceeds the threshold value you have set, the ROs send internal notifications to the RUs and to network administrators in the form of system logging (syslog) messages or Simple Network Management Protocol (SNMP) alerts.

You can set rising and falling values for critical, major, and minor levels of thresholds. When the resource utilization exceeds the rising threshold level, an Up notification is sent. When the resource utilization falls below the falling threshold level, a Down notification is sent.

ERM provides for three types of thresholds to be defined:

- The System Global Threshold is the point when the entire resource reaches a specified value. A notification is sent to all RUs once the threshold is exceeded.
- The User Local Threshold is the point when a specified RUs utilization exceeds the configured limit.
- The User Global Threshold is the point when the entire resource reaches a configured value. A notification is sent to the specified RU once the threshold is exceeded.

System Resources Monitored by the Embedded Resource Manager

ERM monitors CPU, buffer, and memory utilization at a global and task-based level. To avoid infrastructure issues and promote the availability of system resources, the resource owners described in the following sections are monitored:

CPU Resource Owner

The ERM feature uses the existing loadometer process to calculate the load information displayed by the **show processes cpu** command. This method generates a report of the extended load statistics and adds it to a circular buffer every five seconds. You can obtain a record of the load statistics for the past one minute through the CLI. This feature also provides an intelligent CPUHOG profiling mechanism that helps to reduce the time required to diagnose error conditions.

The functions described in the following sections help in load monitoring.

Loadometer Process

The loadometer process generates an extended load monitor report every five seconds. The loadometer function, which calculates process CPU usage percentage, is enhanced to generate the loadometer process reports.

Scheduler

The scheduler collects data when a process is executed, which enables the loadometer to generate reports. The scheduler collects data when the process is launched or when the process transfers control to the scheduler.

Snapshot Management Using Event Trace

Snapshot management manages the buffer in which snapshots of reports are stored. The snapshot management infrastructure stores, displays, and releases the snapshots.

Automatic CPUHOG Profiling

The timer Interrupt Service Routine (ISR) provides automatic CPUHOG profiling. The timer ISR begins profiling a process when it notices that the process has exceeded the configured value or a default of twice the maximum scheduling quantum (maximum time taken for the execution of a task).

On beginning the profiling, the timer ISR saves the interrupted program counter (pc) and return address (ra) in a preallocated buffer. This process provides information that can help the user analyze the CPUHOG.

The profiling continues until the CPUHOG is reported or the buffer is full. To analyze the computation of a long running process you must specify a process ID (PID) and a threshold to start the profiling. When this process takes up more than the specified time (in milliseconds), the profiling begins.

When the data belonging to a particular process exceeds the default size of the buffer, it is reported as a CPUHOG. The default size of the buffer is 1250 entries and can store up to five seconds of profiling data.

Memory Resource Owner

The Embedded Resource Manager feature enhances the memory manager in Cisco IOS devices. The enhancements are described in the following sections:

Memory Usage History

The Embedded Resource Manager feature helps in maintaining memory fragmentation information and thus reduces the need for maintenance of separate scripts for collecting such information.

Memory Accounting

ERM performs the accounting of information for memory by tracking the memory usage of individual RUs. When a process is created, a corresponding RU is also created, against which the usage of memory is recorded. The process of RU creation helps the user to migrate from a process-based accounting to a resource user-based accounting scheme for memory.

The memory RO maintains a global threshold and a per-RU memory usage threshold that can be configured through the ERM infrastructure. The memory RO also tracks the global free memory. When a particular RU's memory usage exceeds the global free memory, a notification is sent to the registered resource monitors

(RMs). Similarly when a particular RU exceeds its threshold of memory usage, a notification is sent to that RU. These notifications are sent using the ERM infrastructure.

A memory RO has the intelligence to assign memory to a RU. When a memory RO receives an allocation request, the memory is assigned to the current RU. When a free request is received, the memory RO reduces the memory assigned to the RU.

Interface Wedging and Packet Memory Leaks

In certain situations, errors in the system accounting of incoming packets can occur, leading to a "memory leak" caused by the input queue. When there is a leak in an interface's input queue, gradually the queue reaches its maximum permitted value, causing the interface to become "wedged." A wedged interface may no longer process incoming packets. Packet memory leaks can cause interface input queue wedges.

The Packet Memory Reclamation functionality improves the infrastructure for preventing wedged interface input queues, and it provides a method for changing the defaults of that infrastructure. The Embedded Resource Manager provides the Packet Memory Reclamation functionality for "unwedging" interface input queues and configuring the system to detect and rectify packet leaks.

Note To use the Packet Memory Reclamation functionality, you must be running Cisco IOS Release 12.4(6)T or a later release. Additional troubleshooting (debugging) commands were introduced by this enhancement for use by technical support representatives in specific situations.

Memory Resource Reclamation for Interfaces

The Garbage Detection process works in conjunction with the Memory RO in achieving interface unwedging (for more details, see the *Memory Leak Dectector* fe ature guide that is part of the *Cisco IOS Configuration Fundamentals Configuration Guide*).

As part of the reclamation process, incoming packets that belong to a leaked input queue can be deallocated and reused. This feature provides a command (**critical rising**) that can be used to fine-tune memory resource reclamation.

Note Configuration of this feature will typically be needed only as part of a troubleshooting process with a Cisco Technical Support representative. Additional configuration tasks or special technical support commands may be required before this feature can be effectively used. Additional **memory debug leak internal service** commands are made available to Cisco Technical Support engineers for use in specific situations.

The deallocation procedure is triggered when a check is made to see if packets are using too much memory. Thresholds for the memory RO can be configured using a global policy of any level.

The purpose of configuring this memory policy is to find a balance between the utilization of the Memory Leak Detector (that can become resource intensive) and the need to detect packet memory leaks. Ideally, the system should perform deallocation only when it becomes absolutely necessary.

The **critical rising** command allows you to set a rising and falling threshold percentage for critical levels of I/O memory usage, and to specify an interval for those values. These values trigger the Memory Leak Detector process and, if needed, the deallocation procedure.

For example, if memory usage is more than that of the rising threshold of 75 percent of total I/O memory for more than 5 seconds, the “critical” notification is generated within the system and a callback is issued. As an action in the callback, a check is made to see if the packets are using too much memory. When the packets have used too much memory, the deallocation procedure begins. If the deallocation procedure does not bring memory utilization below the lower threshold value, the deallocation procedure is periodically reattempted. Once the memory usage falls below the configured threshold value, the periodic attempts to deallocate are stopped.

Memory Leak Reclamation

The Packet Memory Reclamation feature uses the ERM infrastructure to clean up and reclaim leaked Cisco IOS packet memory.

This feature uses the Memory Leak Detector process (sometimes referred to as the Garbage Detection or GD process) and the memory-manager RO functionality to reclaim packet memory.

I O Memory

The I/O memory pool is one of the memory types in Cisco IOS software. The input queue buffers use memory from this pool for processing.

Buffer Resource Owner

The Embedded Resource Manager feature addresses the recurring problems of the Buffer Manager described in the following sections.

Automatic Buffer Tuning

The Embedded Resource Manager feature allows you to automatically tune the buffers using the **buffer tune automatic** command. The buffer RO tunes permanent memory in particle pools based on the usage of the buffer pool.

The buffer RO tracks the number of failures and the availability of memory in the buffer pool. When the number of failures increases above 1 percent of the buffer hits or when no memory is available in the buffer pool, the buffer RO performs an automatic tuning.

Note Ensure that there is sufficient free I/O memory or main memory using the first lines of the **show memory** command output before enabling automatic tuning of buffers.

Here are some keywords from the **buffer tune** command that can help you verify if you have sufficient I/O memory:

- **permanent** : take the number of total buffers in a pool and add 20 percent.
- **min-free** : set the **min-free**keyword to 20 to 30 percent of the permanent number of allocated buffers in the pool.
- **max-free** : set the **max-free**keyword to a value greater than the sum of permanent and minimum values.

However, when there is a traffic burst, the Cisco IOS device may not have enough time to create the new buffers and the number of failures may continue to increase.

The Embedded Resource Manager feature monitors the buffer pool every minute for tuning (that is, for number of hits, number of failures, and the number of counters created). When buffer tuning is enabled, the buffer RO automatically tunes the buffers when required.

Buffer Leak Detection

The Embedded Resource Manager feature allows Cisco IOS devices to detect and diagnose potential buffer leaks. All the buffers in a pool are linked so that they can be traced easily. The number of buffers allocated for incoming and outgoing packets in each buffer pool is tracked and can be displayed in the **show buffers leak** command output.

Buffer Accounting

The Embedded Resource Manager feature consists of mechanisms to account for the usage of buffers. All buffers are owned by the pool manager process (buffer RU). When a RU requests a buffer, the allocated buffer is allotted to that RU. When the RU returns the buffer, it is deducted from the RU's account. The packet type from the output of the **show buffers usage** command indicates the RU to which the packet belongs.

Buffer Usage Thresholding

The Embedded Resource Manager feature provides a facility to manage high buffer utilization. The buffer manager RO registers as a RU with the memory RO. The buffer manager RU is set before a memory allocation is made for creating new buffers. The buffer manager also registers as an RO. When a buffer is allocated, the current RU (if any) is charged with the memory allocation. The buffer manager RO registers for the notifications from the memory manager for the processor and I/O memory pool. If the I/O memory pool is falling short of memory, the buffer manager tries to free the lists of all the buffer pools. If your Cisco IOS device does not support I/O memory, then it registers for notifications from the processor memory.

Cisco IOS software maintains a threshold per buffer pool. When a particular pool exceeds the specified threshold, ERM sends a notification to all the RUs in that pool, so that the RUs can take corrective measures. Thresholds are configured for public buffer pools only.

Global notification is set for every pool in the system; that is, one notification for all pools in the public pool and one notification for each pool in the private pool. Threshold notifications are sent to only those RUs that have registered with the ROs for getting notifications. A list of RUs that have registered with the RO is maintained by the RO. When the threshold of a particular RU is exceeded, then that RU is notified and marked notified. When the buffers are recovered, the notified RUs are moved back to the original list.

For example, an Ethernet driver RU is allocated buffers from some particular private pool. Another RU, Inter Processor Communication (IPC), is added to the list. In this case, when the pool runs low on buffers, the IPC RU gets a notification and it can take corrective measures.

You can configure threshold values as percentages of the total buffers available in the public pool. Total buffer is the sum of maximum allowed buffers and the permanent pools in the public buffer pool. If these values change due to buffer tuning, then the threshold values also change. For example, if the configuration requires that a notification be sent when the IPC RU is holding more than 40 percent of Ethernet buffers and the sum of permanent and maximum allowed for Ethernet buffers is 150 percent, then the Ethernet pool is notified when the IPC RU is holding 60 percent.

Resource Policy Templates

Resource owner policy is a template used by the ROs to associate a RU with a set of thresholds that are configured through the CLI. This template can be used to specify system global, user local, and per user global thresholds. A particular resource group or RU can have only one policy associated with it. The policy template for ROs is maintained by the ERM framework.

When a policy template is associated with a user type and its instance (RUs), the thresholds configured in that policy are applied based on the RU to RO relationship. This method ignores any RO configuration that may not be applicable to the RU.

How to Configure Embedded Resource Manager

Managing Resource Utilization by Defining Resource Policy

Perform this task to configure a resource policy for ERM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]

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.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.

	Command or Action	Purpose
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Enters ERM policy configuration mode to configure a resource policy. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Setting Expected Operating Ranges for Buffer Resources

Perform this task to configure threshold values for buffer RO.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. Do one of the following:
   - **system**
   -
   -
   - **slot** *slot-number*
6. **buffer public**
7. Do one of the following:
   - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   -
   - **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   -
   - **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**

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.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.

	Command or Action	Purpose
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	Do one of the following:
	• **system**
	•
	•
	• **slot** *slot-number*
	
	Example:
	`Router(config-erm-policy)# system`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the Route Switch Processor (RSP).

Step 6	**buffer public** **Example:** `Router(config-policy-node)# buffer public`	Enters buffer owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.
Step 7	Do one of the following: • **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]	Allows you to set the rising and falling threshold values for critical, major, and minor levels of buffer usage count for the public buffer pools.

	Command or Action	Purpose

	•
	• **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	•
	• **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	
	Example:
	`Router(config-owner-buffer)# critical rising 40 falling 20 interval 10 global`
	
	Example:
	
	Example:
	`Router(config-owner-buffer)# major rising 30 falling 15 interval 10 global`
	
	Example:
	
	Example:
	`Router(config-owner-buffer)# minor rising 20 falling 10 interval 10 global`
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global**keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.

Step 8	**exit** **Example:** `Router(config-owner-buffer)# exit`	Exits buffer owner configuration mode.

Setting Expected Operating Ranges for CPU Resources

Perform this task to configure threshold values for the CPU RO.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. Do one of the following:
   - **system**
   -
   -
   - **slot** *slot-number*
6. **cpu interrupt**
7. Do one of the following:
   - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
   -
   - **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
   -
   - **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
8. **exit**
9. **cpu process**
10. Do one of the following:
    - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    -
    - **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    -
    - **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
11. **exit**
12. **cpu total**
13. Do one of the following:
    - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
    -

- **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**
-
- **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global**

14. ex i t

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.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.
Step 5	Do one of the following: • **system** • • • **slot** *slot-number*	Enters policy node configuration mode with the **system** command. Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the RSP.

	Command or Action	Purpose

	Example:
	`Router(config-erm-policy)# system`
	Example:
	Example:
	Example:
	`Router(config-erm-policy)# slot 1`	

Step 6	**cpu interrupt** **Example:** `Router(config-policy-node)# cpu interrupt`	(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.
Step 7	Do one of the following: • **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** • • **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** • • **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** **Example:** `Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global` **Example:**	Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of CPU interrupt utilization. **Note** If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold. For interrupt CPU utilization, you can configure either global thresholds or per user global thresholds. Hence, you must enter the **global** keyword either in Step 4 or in Step 7.

	Command or Action	Purpose

	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	Example:
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`	

Step 8	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits the CPU owner configuration mode.
Step 9	**cpu process** **Example:** `Router(config-policy-node)# cpu process`	(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.
Step 10	Do one of the following: • **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**] • • **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**] • • **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**] **Example:** `Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global` **Example:**	Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of process CPU utilization. **Note** If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 10. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 10 if you want to configure a per user global threshold. For process CPU utilization, you can configure global thresholds, per user global thresholds or user local thresholds.

	Command or Action	Purpose

	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	Example:
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`	

Step 11	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits the CPU owner configuration mode.
Step 12	**cpu total** **Example:** `Router(config-policy-node)# cpu total`	(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.
Step 13	Do one of the following: • **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** • • **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** • • **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] **global** **Example:** `Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global` **Example:**	Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of total CPU utilization. **Note** If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 13. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 13 if you want to configure a per user global threshold. For total CPU utilization, you can configure either global thresholds or per user global thresholds. Hence, you must enter the **global** keyword either in Step 4 or in Step 13.

	Command or Action	Purpose

	Example:
	`Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global`
	Example:
	Example:
	`Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global`	

Step 14	**exit** **Example:** `Router(config-owner-cpu)# exit`	Exits CPU owner configuration mode.

Setting Expected Operating Ranges for Memory Resources

Perform this task to configure threshold values for the memory RO.

Note When the Packet Memory Reclamation functionality is enabled, and the violation of the configured threshold value for the memory RO occurs, the system verifies whether the memory is hogged by the buffers. If 70 percent of the memory is used by the buffers, the system activates the Memory Leak Detector process (sometimes referred to as the “Garbage Detection” or “GD” process) to clean up the memory. (For more details, see the Memory Leak Dectector feature guide that is part of the *Cisco IOS Configuration Fundamentals Configuration Guide*).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. Do one of the following:
   - **system**
   -
   - **slot** *slot-number*
6. **memory io**
7. Do one of the following:
   - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   -
   - **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
   -
   - **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**
9. **memory processor**
10. Do one of the following:
    - **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    -
    - **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
    -
    - **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	Do one of the following:
	• **system**
	•
	• **slot** *slot-number*
	Example:
	`Router(config-erm-policy)# system`
	Example:
	Example:
	`Router(config-erm-policy)# slot 1`
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like the RSP.

	Command or Action	Purpose
	Example:	
Step 6	**memory io** **Example:** `Router(config-policy-node)# memory io`	(Optional) Enters memory owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

Step 7	Do one of the following:
	
	• **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	•
	• **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	•
	• **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	
	Example:
	
	`Router(config-owner-memory)# critical rising 40 falling 20 interval 10 global`
	
	Example:
	
	Example:
	
	`Router(config-owner-memory)# major rising 30 falling 15 interval 10 global`
	
	Example:
	
	Example:
	
	`Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of I/O memory usage.
		
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.
Step 9	**memory processor** **Example:** `Router(config-policy-node)# memory processor`	(Optional) Enters memory owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

Step 10	Do one of the following:
	• **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	•
	• **major rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	•
	• **minor rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
	
	Example:
	`Router(config-owner-memory)# critical rising 40 falling 20 interval 10 global`
	
	Example:
	
	Example:
	`Router(config-owner-memory)# major rising 30 falling 15 interval 10 global`
	
	Example:
	
	Example:
	`Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global`
		Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of processor memory usage.
		Note If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 10. However, if you have configured a user local or per user global policy (by not specifying the **global** keyword) in Step 4, enter the global keyword in Step 10 if you want to configure a per user global threshold.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.

Enabling Automatic Tuning of Buffers

Perform this task to enable automatic tuning of buffers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **buffer tune automatic**

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.
Step 3	**buffer tune automatic** **Example:** `Router(config)# buffer tune automatic`	Enables automatic tuning of buffers.

Managing Memory Usage History

Perform this task to change the number of hours for which the memory log is maintained.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **memory statistics history table** *number-of-hours*

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.
Step 3	**memory statistics history table** *number-of-hours* **Example:** `Router(config)# memory statistics history table 48`	Changes the time (number of hours) for which the memory log is maintained.

Configuring a CPU Process to Be Included in the Extended Load Monitor Report

Perform this task to configure a process (or processes) to be included in the extended load monitor report.

SUMMARY STEPS

1. **enable**
2. **monitor processes cpu extended** *process-id-list*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**monitor processes cpu extended** *process-id-list* **Example:** `Router# monitor processes cpu extended 1`	Enables the specified process or processes to be monitored for the extended CPU load. You can specify a maximum of eight processes to be monitored.

Managing Extended CPU Load Monitoring

Perform this task to change the history size in the collection report for extended CPU load.

Note You cannot disable this feature completely. If the command is not configured, the default behavior is to collect a one-minute history. The one-minute history is equivalent to collecting history for a history size 12.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **process cpu extended history** *history-size*

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.
Step 3	**process cpu extended history** *history-size* **Example:** `Router(config)# process cpu extended history 24`	Enables you to change the history size of the extended collection report. If the command is not configured, the default behavior is to collect a one-minute history, which is equivalent to collecting history for history size 12.

	Command or Action	Purpose

Managing Automatic CPUHOG Profiling

Perform this task to enable automatic profiling of CPUHOGs by the CPU Resource Owner. The CPU Resource Owner predicts when a process could hog CPU and begins profiling that process at the same time. This function is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **processes cpu autoprofile hog**

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.
Step 3	**processes cpu autoprofile hog** **Example:** `Router(config)# processes cpu autoprofile hog`	Enables automatic profiling of CPUHOG processes. This function is enabled by default.

Applying a Policy to Resource Users

Perform this task to apply a policy or policy template to RUs or resource groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. **exit**
6. **user** {*resource-instance-name resource-user-type resource-policy-name* | **global** *global-policy-name* | **group** *resource-group-name* **type** *resource-user-type*}
7. **instance** *instance-name*
8. **policy** *policy-name*

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.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy. • The **type** keyword indicates that you are configuring either a user local or per user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.
Step 5	**exit** **Example:** `Router(config-erm)# exit`	Exits ERM policy configuration mode.

	Command or Action	Purpose
Step 6	**user** {*resource-instance-name resource-user-type resource-policy-name* | **global** *global-policy-name* | **group** *resource-group-name* **type** *resource-user-type*} **Example:** `Router(config-erm)# user group lowPrioUsers type iosprocess` **Example:**	Applies a policy system wide (global thresholding), a group of users (group thresholding), or a particular user. **Note** When you apply a group policy to a group of RUs by giving the **group** keyword in this command, the Cisco IOS router enters the resource group configuration mode. Go to Step 7 if you want to add RUs to the resource group. Got to Step 8 if you want to apply a policy to the resource group. • The *resource-instance-name* argument identifies the name of the RU to which you are applying a policy. • The *resource-user-type-name* argument identifies the type of RU. • The *resource-policy-name* argument identifies the name resource policy you are applying to the individual RU. • The *global-policy-name* argument identifies the name of the global policy you are trying to apply. • The *resource-group-name* argument identifies the name of the resource group.
Step 7	**instance** *instance-name* **Example:** `Router(config-res-group)# instance http`	Adds an RU to a resource group. The *instance-name* argument specifies the RU or instance name. **Note** All the RUs added by this command will be grouped together under the resource group and the same thresholding policy will be applied to all the RUs. For example, if you have created a resource group lowPrioUsers in Step 6, then all the RUs you add in Step 7 will be part of the resource group lowPrioUsers and the same policy is applied to all the RUs.
Step 8	**policy** *policy-name* **Example:** `Router(config-res-group)# policy group-policy1`	Specifies the policy you want to apply to the resource group you created in Step 6. The *policy-name* argument specifies the name of the group policy. This command helps you to set the same threshold policy to a group of RUs grouped under a resource group. For example, if you have some low-priority tasks or RUs like **http** and **snmp** and you want to set a threshold not on these individual RUs, but as a group; then add these RUs to the lowPrioUsers group using Step 7 and then apply a threshold policy using Step 8. In this case, if you have set a minor rising threshold of 10 percent (this 10 percent threshold is applied to both **http** and **snmp** in the lowPrioUsers group), then a notification is sent to lowPrioUsers resource group when the accumulated usage exceeds the 10 percent mark. That is, if http uses 4 percent and snmp uses 7 percent, a notification will be sent to all the RUs in the lowPrioUsers resource group.

Setting a Critical Rising Threshold for Global I O Memory

Perform this task to specify a critical rising threshold value for the global I/O memory pool. If global I/O memory resource consumption meets or exceeds this value, the Memory Leak Detector process will be

automatically triggered. This configuration is only needed if you are experiencing a problem and you want to change (fine tune) how often the automatic process occurs (for example, set the threshold lower so that deallocation check occurs more frequently).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **resource policy**
4. **policy** *policy-name* [**global** | **type** *resource-user-type*]
5. Do one of the following:
   - **system**
   -
   - **slot** *slot-number*
6. **memory io**
7. **critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**]
8. **exit**

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.
Step 3	**resource policy** **Example:** `Router(config)# resource policy`	Enters ERM configuration mode.
Step 4	**policy** *policy-name* [**global** | **type** *resource-user-type*] **Example:** `Router(config-erm)# policy policy1 type iosprocess`	Configures a resource policy and enters ERM policy configuration mode. • The *policy-name* argument identifies the name of the resource policy. • The **global** keyword is used when you are configuring a system global policy.

	Command or Action	Purpose
		• The **type** keyword indicates that you are configuring either a user local or per-user global policy. The *resource-user-type* argument identifies the name of the resource user type you want to attach the policy to.

Step 5	Do one of the following:
	• **system**
	•
	• **slot** *slot-number*
	Example:
	`Router(config-erm-policy)# system`
	Example:
	Example:
	`Router(config-erm-policy)# slot 1`
	Example:
		Enters policy node configuration mode with the **system** command.
		Enters ERM slot configuration mode with the **slot** *slot-number* command. This command is available only in distributed platforms like RSP.

Step 6	**memory io** **Example:** `Router(config-policy-node)# memory io`	(Optional) Enters memory owner configuration mode. • Allows you to set the rising and falling values for the critical, major, and minor thresholds.
Step 7	**critical rising** *rising-threshold-value* [**interval** *interval-value*] [**falling** *falling-threshold-value* [**interval** *interval-value*]] [**global**] **Example:** `Router(config-owner-memory)# critical rising 75 falling 65 interval 10 global` **Example:**	Allows you to set the rising and falling threshold values for critical levels as percentages of the I/O memory pool, and set the interval of time that must pass before these values are registered. • If the amount of memory held by the resource user exceeds the rising threshold value, a rising threshold notification is generated. • If the falling threshold notification is generated before the interval has passed, then the rising notification is not sent. • The interval following the rising threshold signifies this time period in seconds. • If the amount of memory held by the resource user falls below the falling threshold, the falling threshold notification is sent.

	Command or Action	Purpose
		• The optional **global** keyword indicates that the threshold is being set on the global memory consumption, not on the memory used by the particular resource user in which the configuration is being applied. **Note** If you had configured a global policy in Step 4, you do not need to give the **global** keyword while setting the threshold values in Step 7. However, if you have configured a user local or per-user global policy (by not specifying the **global** keyword) in Step 4, enter the **global** keyword in Step 7 if you want to configure a per user global threshold.
Step 8	**exit** **Example:** `Router(config-owner-memory)# exit`	Exits memory owner configuration mode.

Verifying ERM Operations

To verify the various ERM operations, perform the following steps.

SUMMARY STEPS

1. **show buffers leak** [**resource user**]
2. **show buffers tune**
3. **show buffers usage** [**pool** *pool-name*]
4. **show memory** [**processor** | **io**] **fragment** [**detail**]
5. **show memory statistics history table**
6. **show monitor event-trace cpu-report** {**brief** {**all** [**detail**] | **back** *time* | **clock** *time* | **from-boot** [*seconds* | **detail**] | **latest** [**detail**]} | **handle** *handle-number*}
7. **show processes cpu autoprofile hog**
8. **show processes cpu extended** [**history**]
9. **show resource all** [**brief** | **detailed**]
10. **show resource database**
11. **show resource owner** {*resource-owner-name* | **all**} **user** {*resource-user-type-name* | **all**} [**brief** | **detailed** | **triggers**]
12. **show resource relationship user** *resource-user-type*
13. **show resource user** {**all** | *resource-user-type*} [**brief** | **detailed**]

DETAILED STEPS

Step 1 **show buffers leak** [**resource user**]

Use this command without the optional keywords to display the details of all the buffers that are older than one minute in the system, for example:

Example:

```
Router# show buffers leak
Header   DataArea  Pool  Size  Link  Enc    Flags      Input     Output  User
6488F464  E000084 Small    74     0    0       10       None       None EEM ED Sy
6488FB5C  E000304 Small    74     0    0       10       None       None EEM ED Sy
648905D0  E0006C4 Small    61     0    0        0       None       None EEM ED Sy
648913C0  E000BC4 Small    74     0    0       10       None       None EEM ED Sy
6489173C  E000D04 Small    74     0    0       10       None       None EEM ED Sy
648921B0  E0010C4 Small    60     0    0        0       None       None Init
6489252C  E001204 Small   103     0    0       10       None       None EEM ED Sy
64892C24  E001484 Small    74     0    0       10       None       None EEM ED Sy
64892FA0  E0015C4 Small    74     0    0       10       None       None EEM ED Sy
64893A14  E001984 Small    74     0    0       10       None       None EEM ED Sy
64893D90  E001AC4 Small    61     0    0        0       None       None EEM ED Sy
64894804  E001E84 Small    61     0    0        0       None       None EEM ED Sy
6517CB64  E32F944 Small    74     0    0       10       None       None EEM ED Sy
6517D25C  E176D44 Small    74     0    0       10       None       None EEM ED Sy
6517D5D8  E176E84 Small    74     0    0       10       None       None EEM ED Sy
6517D954  E209A84 Small    74     0    0       10       None       None EEM ED Sy
6517E744  E209D04 Small    61     0    0        0       None       None EEM ED Sy
6517EE3C  E29CBC4 Small    61     0    0        0       None       None EEM ED Sy
65180324  E177844 Small    74     0    0       10       None       None EEM ED Sy
65180D98  E177C04 Small    61     0    0        0       None       None EEM ED Sy
65E1F3A0  E4431A4 Small   102     0    0        0       None       None EEM ED Sy
64895278  E002644 Middl   191     0    0       10       None       None EEM ED Sy
64895CEC  E003004 Middl   173     0    0       10       None       None EEM ED Sy
64896068  E003344 Middl   176     0    0       10       None       None EEM ED Sy
648963E4  E003684 Middl   191     0    0       10       None       None EEM ED Sy
64896E58  E004044 Middl   109     0    0       10       None       None EEM ED Sy
64897C48  E004D44 Middl   194     0    0       10       None       None EEM ED Sy
65181F04  E330844 Middl   173     0    0       10       None       None EEM ED Sy
65183070  E3C3644 Middl   105     0    0       10       None       None EEM ED Sy
65DF9558  E4746E4 Middl   107     0    0        0       None       None EEM ED Sy
65DFA6C4  E475724 Middl   116     0    0        0       None       None EEM ED Sy
65DFADBC  E475DA4 Middl   115     0    0        0       None       None EEM ED Sy
65DFC620  E477464 Middl   110     0    0        0       None       None EEM ED Sy
64C64AE0        0 FS He     0     0    3        0       None       None Init
64C64E5C        0 FS He     0     0    3        0       None       None Init
64C651D8        0 FS He     0     0    3        0       None       None Init
64C65554        0 FS He     0     0    0        0       None       None Init
64C658D0        0 FS He     0     0    0        0       None       None Init
64C65C4C        0 FS He     0     0    0        0       None       None Init
64C65FC8        0 FS He     0     0    0        0       None       None Init
64C66344        0 FS He     0     0    0        0       None       None Init
64D6164C        0 FS He     0     0    0        0       None       None Init
64EB9D10        0 FS He     0     0    0        0       None       None Init
6523EE14        0 FS He     0     0    0        0       None       None Init
65413648        0 FS He     0     0    0        0       None       None Init
```

Use this command with the optional keywords to display the details of the buffers of a specified RU that are older than one minute in the system, for example:

Example:

```
Router# show buffers leak resource user
Resource User:  EEM ED Syslog count:       32
Resource User:           Init count:        2
Resource User:         *Dead* count:        2
```

```
Resource User: IPC Seat Manag count:       11
Resource User:      XDR mcast count:        2
```

Step 2 **show buffers tune**

Use this command to display the details of automatic tuning of buffers, for example:

Example:

```
Router# show buffers tune

Tuning happened for the pool Small
Tuning happened at 20:47:25
Oldvalues
permanent:50  minfree:20  maxfree:150
Newvalues
permanet:61  minfree:15  maxfree:76
Tuning happened for the pool Middle
Tuning happened at 20:47:25
Oldvalues
permanent:25  minfree:10  maxfree:150
Newvalues
permanet:36  minfree:9  maxfree:45
```

Step 3 **show buffers usage** [**pool** *pool-name*]

Use this command without the optional keyword and argument to display the details of the buffer usage pattern in a specified buffer pool, for example:

Example:

```
Router# show buffers usage

Statistics for the Small pool
Caller pc    : 0x626BA9E0 count:       20
Resource User: EEM ED Sys count:       20
Caller pc    : 0x60C71F8C count:        1
Resource User:       Init count:        1
Number of Buffers used by packets generated by system:   62
Number of Buffers used by incoming packets:               0
Statistics for the Middle pool
Caller pc    : 0x626BA9E0 count:       12
Resource User: EEM ED Sys count:       12
Number of Buffers used by packets generated by system:   41
Number of Buffers used by incoming packets:               0
Statistics for the Big pool
Number of Buffers used by packets generated by system:   50
Number of Buffers used by incoming packets:               0
Statistics for the VeryBig pool
Number of Buffers used by packets generated by system:   10
Number of Buffers used by incoming packets:               0
Statistics for the Large pool
Number of Buffers used by packets generated by system:    0
Number of Buffers used by incoming packets:               0
Statistics for the Huge pool
Number of Buffers used by packets generated by system:    0
Number of Buffers used by incoming packets:               0
Statistics for the IPC pool
Number of Buffers used by packets generated by system:    2
Number of Buffers used by incoming packets:               0
Statistics for the Header pool
Number of Buffers used by packets generated by system:  511
Number of Buffers used by incoming packets:               0
Statistics for the FS Header pool
Caller pc    : 0x608F68FC count:        9
Resource User:       Init count:       12
Caller pc    : 0x61A21D3C count:        1
Caller pc    : 0x60643FF8 count:        1
```

```
Caller pc    : 0x61C526C4 count:        1
Number of Buffers used by packets generated by system:   28
Number of Buffers used by incoming packets:               0
```

Use this command with the optional keyword and argument to display the details of the buffer usage pattern in a small buffer pool, for example:

Example:

```
Router# show buffers usage pool small
Statistics for the Small pool
Caller pc    : 0x626BA9E0 count:       20
Resource User: EEM ED Sys count:       20
Caller pc    : 0x60C71F8C count:        1
Resource User:       Init count:        1
Number of Buffers used by packets generated by system:   62
Number of Buffers used by incoming packets:               0
```

Step 4 **show memory** [**processor** | **io**] **fragment** [**detail**]

Use this command without the optional keywords to display the block details of every allocated block for both I/O memory and processor memory, for example:

Example:

```
Router# show memory fragment

Processor memory
Free memory size : 211014448 Number of free blocks:      139
Allocator PC Summary for allocated blocks in pool: Processor
    PC          Total     Count  Name
0x6189A438     318520       1  RTPSPI
0x6205711C     237024       2  CCH323_CT
0x6080BE38      98416       2  Exec
0x606AD988      80256       1  Init
0x618F68A8      73784       1  CCSIP_UDP_SOCKET
0x6195AD04      67640       1  QOS_MODULE_MAIN
0x606488C8      65592       1  CEF: Adjacency chunk
0x60635620      65592       1  CEF: 16 path chunk pool
0x615ECE58      65592       1  XTagATM VC chunk
0x6165ACF8      65592       1  eddri_self_event
0x608DE168      65592       1  MallocLite
0x60857920      51020      11  Normal
0x6203BF88      42480       4  IPv6 CEF fib tables
0x60DC7F14      32824       1  PPP Context Chunks
.
.
.
I/O memory
Free memory size : 14700024 Number of free blocks:       52
Allocator PC Summary for allocated blocks in pool: I/O
    PC          Total   Count   Name
0x60857934    3936000      60  FastEthernet0/
0x60857898     524800       8  FastEthernet0/0
0x601263CC      29120       7  Init
0x6082DB28       9408      23  *Packet Data*
0x60126344       8448       4  Init
Allocator PC Summary for free blocks in pool: I/O
    PC          Total    Count   Name
0x608C5730   29391444       1  (coalesced)
0x608FC1F4       5376      28  (fragment)
0x6082DB28       4288      14  (fragment)
```

Use this command with the **detail** optional keyword to display the block details of every allocated block for both I/O memory and processor memory, for example:

Example:

```
Router# show memory fragment detail

Processor memory
Free memory size : 211038812 Number of free blocks:      139
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
644AAB70 0000001032 644AAB20 644AAFAC 001  -------- -------- 620450F8  Index Table Block
644AAFAC 0000000028 644AAB70 644AAFFC 000  0        6448CB5C 607B2ADC  NameDB String
644AAFFC 0000000076 644AAFAC 644AB07C 001  -------- -------- 60818DE0  Init
6448CB0C 0000000028 6448CABC 6448CB5C 001  -------- -------- 607F8380  Cond Debug definition
6448CB5C 0000000028 6448CB0C 6448CBAC 000  644AAFAC 6489F158 607B2ADC  NameDB String
6448CBAC 0000000028 6448CB5C 6448CBFC 001  -------- -------- 607F8380  Cond Debug definition
6489EF8C 0000000408 6489DBCC 6489F158 001  -------- -------- 60857920  Normal
6489F158 0000000064 6489EF8C 6489F1CC 000  6448CB5C 6448CABC 607B2ADC  NameDB String
6489F1CC 0000005004 6489F158 648A058C 001  -------- -------- 60857920  Normal
6448CA6C 0000000028 6448C9AC 6448CABC 001  -------- -------- 607D72FC  Parser Linkage
6448CABC 0000000028 6448CA6C 6448CB0C 000  6489F158 644949C8 607B2ADC  NameDB String
6448CB0C 0000000028 6448CABC 6448CB5C 001  -------- -------- 607F8380  Cond Debug definition
64494978 0000000028 64494928 644949C8 001  -------- -------- 607D72FC  Parser Linkage
644949C8 0000000028 64494978 64494A18 000  6448CABC 654F2868 607B2ADC  NameDB String
64494A18 0000000028 644949C8 64494A68 001  -------- -------- 607D72FC  Parser Linkage
654F27E8 0000000076 654F2768 654F2868 001  -------- -------- 60818DE0  Init
654F2868 0000000076 654F27E8 654F28E8 000  644949C8 654F1BE8 60818DE0  Init
.
.
.
I/O memory
Free memory size : 14700024 Number of free blocks:       52
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
0E000000 0000000056 00000000 0E00006C 000  0        E176F4C  00000000  (fragment)
0E00006C 0000000268 0E000000 0E0001AC 001  -------- -------- 6082DB28  *Packet Data*
0E176E0C 0000000268 0E176CCC 0E176F4C 001  -------- -------- 6082DB28  *Packet Data*
0E176F4C 0000000076 0E176E0C 0E176FCC 000  E000000  E209F4C  6082DB28  (fragment)
0E176FCC 0000002060 0E176F4C 0E17780C 001  -------- -------- 60126344  Init
0E209E0C 0000000268 0E209CCC 0E209F4C 001  -------- -------- 6082DB28  *Packet Data*
0E209F4C 0000000076 0E209E0C 0E209FCC 000  E176F4C  E29CF4C  6082DB28  (fragment)
0E209FCC 0000002060 0E209F4C 0E20A80C 001  -------- -------- 60126344  Init
0E29CE0C 0000000268 0E29CCCC 0E29CF4C 001  -------- -------- 6082DB28  *Packet Data*
0E29CF4C 0000000076 0E29CE0C 0E29CFCC 000  E209F4C  E32FF4C  6082DB28  (fragment)
0E29CFCC 0000002060 0E29CF4C 0E29D80C 001  -------- -------- 60126344  Init
0E32FE0C 0000000268 0E32FCCC 0E32FF4C 001  -------- -------- 6082DB28  *Packet Data*
0E32FF4C 0000000076 0E32FE0C 0E32FFCC 000  E29CF4C  0        6082DB28  (fragment)
0E32FFCC 0000002060 0E32FF4C 0E33080C 001  -------- -------- 60126344  Init
0E177FCC 0000004108 0E177E4C 0E17900C 001  -------- -------- 601263CC  Init
0E17900C 0000000140 0E177FCC 0E1790CC 000  0        E18910C  601263CC  (fragment)
```

Use this command with **detail** optional keyword to display the block details of every allocated block for processor memory, for example:

Example:

```
Router# show memory processor fragment detail

Processor memory
Free memory size : 65566148 Number of free blocks:      230
 Address      Bytes     Prev     Next Ref     PrevF    NextF Alloc PC  what
645A8148 0000000028 645A80F0 645A8194 001  -------- -------- 60695B20  Init
645A8194 0000000040 645A8148 645A81EC 000  0        200B4300 606B9614  NameDB String
645A81EC 0000000260 645A8194 645A8320 001  -------- -------- 607C2D20  Init
200B42B4 0000000028 200B4268 200B4300 001  -------- -------- 62366C80  Init
200B4300 0000000028 200B42B4 200B434C 000  645A8194 6490F7E8 60976574  AAA Event Data
200B434C 0000002004 200B4300 200B4B50 001  -------- -------- 6267D294  Coproc Request Structures
6490F79C 0000000028 6490F748 6490F7E8 001  -------- -------- 606DDA04  Parser Linkage
6490F7E8 0000000028 6490F79C 6490F834 000  200B4300 6491120C 606DD8D8  Init
6490F834 0000006004 6490F7E8 64910FD8 001  -------- -------- 607DF5BC  Process Stack
649111A0 0000000060 64911154 6491120C 001  -------- -------- 606DE82C  Parser Mode
```

```
6491120C 0000000028 649111A0 64911258 000  6490F7E8 500770F0 606DD8D8  Init
64911258 0000000200 6491120C 64911350 001  -------- -------- 603F0E38  Init
.
20000000 0000000828 5C3AEB24 2000036C 001  -------- -------- 60734010  *Packet Header*
6500BF94 0000000828 6500BC28 6500C300 001  -------- -------- 60734010  *Packet Header*
6500C300 0004760912 6500BF94 50000000 000  5C3AEB24 2C42E310 6071253C  (coalesced)
50000000 0000000828 6500C300 5000036C 001  -------- -------- 60734010  *Packet Header*
2C42E0B4 0000000556 2C429430 2C42E310 001  -------- -------- 60D4A0B4  Virtual Exec
2C42E310 0062725312 2C42E0B4 00000000 000  6500C300 0        6071253C  (coalesced)
```

Use this command with **detail** optional keyword to display the block details of every allocated block for I/O memory, for example:

Example:

```
Router# show memory io fragment detail

0E3F8BAC 0000000204 0E3F8AAC 0E3F8CAC 001  -------- -------- 608C5730  test memory
0E3F8CAC 0000000204 0E3F8BAC 0E3F8DAC 000  0        E3F8AAC  608C5730  test memory
0E3F8DAC 0000000204 0E3F8CAC 0E3F8EAC 001  -------- -------- 608C5730  test memory
0E3F89AC 0000000204 0E3F88AC 0E3F8AAC 001  -------- -------- 608C5730  test memory
0E3F8AAC 0000000204 0E3F89AC 0E3F8BAC 000  E3F8CAC  E3F88AC  608C5730  test memory
0E3F8BAC 0000000204 0E3F8AAC 0E3F8CAC 001  -------- -------- 608C5730  test memory
0E3F87AC 0000000204 0E3F86AC 0E3F88AC 001  -------- -------- 608C5730  test memory
0E3F88AC 0000000204 0E3F87AC 0E3F89AC 000  E3F8AAC  E3F86AC  608C5730  test memory
0E3F89AC 0000000204 0E3F88AC 0E3F8AAC 001  -------- -------- 608C5730  test memory
0E3F85AC 0000000204 0E3F826C 0E3F86AC 001  -------- -------- 608C5730  test memory
0E3F86AC 0000000204 0E3F85AC 0E3F87AC 000  E3F88AC  0        608C5730  test memory
0E3F87AC 0000000204 0E3F86AC 0E3F88AC 001  -------- -------- 608C5730  test memory
0E3F4E6C 0000000268 0E3F4D2C 0E3F4FAC 000  0        E3F5BEC  608C5730  test memory
0E3F5BEC 0000000268 0E3F5AAC 0E3F5D2C 000  E3F4E6C  E3EE56C  608C5730  test memory
0E3EE46C 0000000204 0E3EE12C 0E3EE56C 001  -------- -------- 608C5730  test memory
0E3EEFAC 0000000204 0E3EEE6C 0E3EF0AC 001  -------- -------- 608C5730  test memory
0E3F06EC 0000000204 0E3F03AC 0E3F07EC 001  -------- -------- 608C5730  test memory
0E3F8DAC 0000000204 0E3F8CAC 0E3F8EAC 001  -------- -------- 608C5730  test memory
```

Step 5 **show memory statistics history table**

Use this command to display the history of memory consumption, for example:

Example:

```
Router# show memory statistics history table

History for Processor memory
Time: 15:48:56.806
Used(b): 422748036 Largest(b): 381064952 Free blocks :291
Maximum memory users for this period
Process Name           Holding   Num Alloc
Virtual Exec             26992          37
TCP Protocols            14460           6
IP Input                  1212           1
Time: 14:42:54.506
Used(b): 422705876 Largest(b): 381064952 Free blocks :296
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                  400012740          24
Dead                    1753456          90
Pool Manager             212796         257
Time: 13:37:26.918
Used(b): 20700520 Largest(b): 381064952 Free blocks :196
Maximum memory users for this period
Process Name           Holding   Num Alloc
Exec                       8372           5
Time: 12:39:44.422
Used(b): 20701436 Largest(b): 381064952 Free blocks :193
Time: 11:46:25.135
```

```
Used(b): 20701436 Largest(b): 381064952 Free blocks :193
Maximum memory users for this period
Process Name           Holding   Num Alloc
CDP Protocol              3752          25
.
.
.
History for I/O memory
Time: 15:48:56.809
Used(b):  7455520 Largest(b): 59370080 Free blocks :164
Time: 14:42:54.508
Used(b):  7458064 Largest(b): 59370080 Free blocks :165
Maximum memory users for this period
Process Name           Holding   Num Alloc
Pool Manager            141584         257
Time: 13:37:26.920
Used(b):  7297744 Largest(b): 59797664 Free blocks :25
Time: 12:39:44.424
Used(b):  7297744 Largest(b): 59797664 Free blocks :25
.
.
.
Time: 09:38:53.040
Used(b):  7297744 Largest(b): 59797664 Free blocks :25
Time: 01:02:05.533
Used(b):  7308336 Largest(b): 59797664 Free blocks :23
Time: 00:00:17.937
Used(b):  7308336 Largest(b): 59797664 Free blocks :23
Maximum memory users for this period
Process Name           Holding   Num Alloc
Init                   7296000         214
Pool Manager               816           3
```

Step 6 **show monitor event-trace cpu-report** {**brief** {**all** [**detail**] | **back** *time* | **clock** *time* | **from-boot** [*seconds* | **detail**] | **latest** [**detail**]} | **handle** *handle-number*}

Use this command to view a brief CPU report details for event tracing on a networking device, for example:

Example:

```
Router# show monitor event-trace cpu-report brief all
Timestamp   : Handle Name              Description
00:01:07.320:  1     CPU               None
```

Use this command to view a brief CPU report details for event tracing on a networking device, for example:

Example:

```
Router# show monitor event-trace cpu-report handle 1
00:01:07.320:  1     CPU               None
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:03:56
Queue Statistics
----------------
             Exec Count    Total CPU  Response Time       Queue Length
                                         (avg/max)          (avg/max)
Critical           1          0          0/0                1/1
High               5          0          0/0                1/1
Normal           178          0          0/0                2/9
Low               15          0          0/0                2/3
Common Process Information
------------------------------
 PID   Name         Prio Style
------------------------------
  10 AAA high-capacit M  New
```

```
 133 RADIUS TEST CMD  M  New
  47 VNM DSPRM MAIN   H  New
  58 TurboACL         M  New
  97 IP Background    M  New
  99 CEF: IPv4 proces L  New
 112 X.25 Background  M  New
 117 LFDp Input Proc  M  New
   3 Init             M  Old
CPU Intensive processes
------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall  Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)      Count Per avg/max
------------------------------------------------------------------------------
   3   820          6   136/236         1     24/24              18  887/15172
Priority Suspends
-----------------------------------
 PID Exec Count Prio-Susps
-----------------------------------
   3          6          1
Latencies
-------------------------
 PID     Exec Count Latency
                    avg/max
-------------------------
  10          1 15192/15192
 133          1 15192/15192
  58          1 15192/15192
 112          1 15192/15192
 117          1 15192/15192
  99          1 15172/15172
  47          1 15172/15172
  97          1 15172/15172
################################################################################
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 00:00:00
Queue Statistics
----------------
         Exec Count  Total CPU    Response Time           Queue Length
                                    (avg/max)               (avg/max)
Critical      0          0             0/0                    0/0
High          0          0             0/0                    0/0
Normal        0          0             0/0                    0/0
Low           0          0             0/0                    0/0

Common Process Information
------------------------------
 PID Name            Prio Style
------------------------------
CPU Intensive processes
------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall  Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)      Count Per avg/max
------------------------------------------------------------------------------
Priority Suspends
-----------------------------------
 PID Exec Count Prio-Susps
-----------------------------------
Latencies
-------------------------
 PID Exec Count   Latency
                  avg/max
-------------------------
################################################################################
```

Step 7 **show processes cpu autoprofile hog**

Use this command to view the CPUHOG autoprofile data, for example:

Example:

```
Router# show processes cpu autoprofile hog
0x6075DD40 0x60755638
0x6075DD24 0x60755638
0x6075563C 0x60755638
0x60755638 0x60755638
0x60755638 0x60755638
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075DD40 0x60755638
0x6075563C 0x60755638
0x6075DCE0 0x60755638
0x6075DD44 0x60755638
0x6075DCCC 0x60755638
0x6075DD10 0x60755638
.
.
.
0x6075DD3C 0x60755638
0x6075DD38 0x60755638
0x6075DD10 0x60755638
0x6075DCCC 0x60755638
0x6075DCDC 0x60755638
0x6075563C 0x60755638
0x6075DD3C 0x60755638
0x6075DD20 0x60755638
0x6075DD58 0x60755638
0x6075DD1C 0x60755638
0x6075DD10 0x60755638
0x6075DCDC 0x60755638
0x6075DCF8 0x60755638
```

Step 8 **show processes cpu extended** [**history**]

Use this command to view an extended CPU load report, for example:

Example:

```
Router# show processes cpu extended
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:03:56
Queue Statistics
----------------
             Exec Count  Total CPU    Response Time         Queue Length
                                       (avg/max)             (avg/max)
Critical           1          0          0/0                   1/1
High               5          0          0/0                   1/1
Normal           178          0          0/0                   2/9
Low               15          0          0/0                   2/3
Common Process Information
------------------------------
 PID Name             Prio Style
------------------------------
CPU Intensive processes
--------------------------------------------------------------------------------
 PID Total       Exec    Quant         Burst  Burst size  Schedcall  Schedcall
     CPUms      Count   avg/max        Count avg/max(ms)      Count Per avg/max
--------------------------------------------------------------------------------
Priority Suspends
-----------------------------------
 PID Exec Count Prio-Susps
-----------------------------------
Latencies
-------------------------
```

```
 PID Exec Count   Latency
                  avg/max
-------------------------
################################################################################
```

Step 9 **show resource all [brief | detailed]**

Use this command without the optional keywords to display the resource details, for example:

Example:

```
Router# show resource all
Resource Owner: cpu
Resource User Type: iosprocess
Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           0        13          0  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223        2872     36029         79  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225      352744     33446      10546  0.00%  0.20%  0.17% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         2          0  0.00%  0.00%  0.00% AAA high-capacit
  Resource User: Policy Manager(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777232           0         1          0  0.00%  0.00%  0.00% Policy Manager
  Resource User: Crash writer(ID: 0x1000011)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777233           0         1          0  0.00%  0.00%  0.00% Crash writer
  Resource User: RO Notify Timers(ID: 0x1000012)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777234           0         1          0  0.00%  0.00%  0.00% RO Notify Timers
  Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777235           0         1          0  0.00%  0.00%  0.00% RMI RM Notify Wa
  Resource User: EnvMon(ID: 0x1000014)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
```

```
16777236       11164     92859        120  0.00%  0.00%  0.00% EnvMon
  Resource User: IPC Dynamic Cache(ID: 0x1000015)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777237           0      3004          0  0.00%  0.00%  0.00% IPC Dynamic Cach
  Resource User: IPC Periodic Timer(ID: 0x1000017)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777239           0    180082          0  0.00%  0.00%  0.00% IPC Periodic Tim
  Resource User: IPC Managed Timer(ID: 0x1000018)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777240         572     79749          7  0.00%  0.00%  0.00% IPC Managed Time
  Resource User: IPC Deferred Port Closure(ID: 0x1000019)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777241           4    180088          0  0.00%  0.00%  0.00% IPC Deferred Por
  Resource User: IPC Seat Manager(ID: 0x100001A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777242       97560   1408799         69  0.23%  0.02%  0.00% IPC Seat Manager
  Resource User: IPC Session Service(ID: 0x100001B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777243           0         1          0  0.00%  0.00%  0.00% IPC Session Serv
  Resource User: ARP Input(ID: 0x100001C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777244          20      3082          6  0.00%  0.00%  0.00% ARP Input
  Resource User: EEM ED Syslog(ID: 0x100001D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777245           0        49          0  0.00%  0.00%  0.00% EEM ED Syslog
  Resource User: DDR Timers(ID: 0x100001E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777246           0         2          0  0.00%  0.00%  0.00% DDR Timers
  Resource User: Dialer event(ID: 0x100001F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777247           0         2          0  0.00%  0.00%  0.00% Dialer event
  Resource User: Entity MIB API(ID: 0x1000020)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777248          28        16       1750  0.00%  0.00%  0.00% Entity MIB API
.
.
.
Resource User: draco-oir-process:slot 2(ID: 0x100011E)
Getbufs  Retbufs  Holding  RU Name
0        0        0        draco-oir-proces
  Resource User: SCP async: Draco-LC4(ID: 0x1000125)
Getbufs  Retbufs  Holding  RU Name
35849    243101   4294760044 SCP async: Draco
  Resource User: IFCOM Msg Hdlr(ID: 0x1000127)
Getbufs  Retbufs  Holding  RU Name
2        2        0        IFCOM Msg Hdlr
  Resource User: IFCOM Msg Hdlr(ID: 0x1000128)
Getbufs  Retbufs  Holding  RU Name
28       28       0        IFCOM Msg Hdlr
  Resource User: Exec(ID: 0x100012C)
Getbufs  Retbufs  Holding  RU Name
912      912      0        Exec
Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
```

Step 10 **show resource database**

Use this command to display the resource database details, for example:

Example:

```
Router# show resource database
List of all Resource Owners :
Owner: cpu                        Id:0x1
Owner's list of monitors is empty.
Owner: memory                     Id:0x2
```

```
Owner's list of monitors is empty.
Owner: Buffer                    Id:0x3
Owner's list of monitors is empty.
Owner: test_mem                  Id:0x4
Owner's list of monitors is empty.
Owner: test_cpu                  Id:0x5
Owner's list of monitors is empty.
Owner: test_RO0                  Id:0x7
Owner's list of monitors is empty.
Owner: test_RO1                  Id:0x8
Owner's list of monitors is empty.
Owner: test_RO2                  Id:0x9
Owner's list of monitors is empty.
Owner: test_RO3                  Id:0xA
Owner's list of monitors is empty.
.
.
.
Resource Monitor: test_ROM0, ID: 0x1B
 Not Watching any Relations.
 Not Watching any Policies.
Resource Monitor: test_ROM1, ID: 0x1C
 Not Watching any Relations.
 Not Watching any Policies.
Resource Monitor: test_ROM2, ID: 0x1D
 Not Watching any Relations.
 Not Watching any Policies.
```

Step 11 **show resource owner** {*resource-owner-name* | **all**} **user** {*resource-user-type-name* | **all**} [**brief** | **detailed** | **triggers**]
Use this command to display the resource owner details, for example:

Example:

```
Router# show resource owner all user all
Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223           4       292         13  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225         376       192       1958  0.00%  0.02%  0.00% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
```

```
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: ALARM_TRIGGER_SCAN(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0       268          0  0.00%  0.00%  0.00% ALARM_TRIGGER_SC
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource User Type: test_RUT143
Resource User Type: test_RUT144
Resource User Type: test_RUT145
Resource User Type: test_RUT146
Resource User Type: test_RUT147
```

Step 12 **show resource relationship user** *resource-user-type*

Use this command to display the relationship details between different resource owners, for example:

Example:

```
Router# show resource relationship
Resource User Type: iosprocess (ID: 0x1)
 -> Resource Owner: cpu (ID: 0x1)
 -> Resource Owner: memory (ID: 0x2)
 -> Resource Owner: Buffer (ID: 0x3)
 -> Resource User: Init (ID: 0x1000001)
 -> Resource User: Scheduler (ID: 0x1000002)
 -> Resource User: Dead (ID: 0x1000003)
 -> Resource User: Interrupt (ID: 0x1000004)
 -> Resource User: Memory RO RU (ID: 0x1000005)
 -> Resource User: Chunk Manager (ID: 0x1000006)
 -> Resource User: Load Meter (ID: 0x1000007)
 -> Resource User: Check heaps (ID: 0x1000009)
 -> Resource User: Pool Manager (ID: 0x100000A)
 -> Resource User: Buffer RO RU (ID: 0x100000B)
 -> Resource User: Timers (ID: 0x100000C)
 -> Resource User: Serial Background (ID: 0x100000D)
 -> Resource User: ALARM_TRIGGER_SCAN (ID: 0x100000E)
 -> Resource User: AAA_SERVER_DEADTIME (ID: 0x100000F)
 -> Resource User: AAA high-capacity counters (ID: 0x1000010)
 -> Resource User: Policy Manager (ID: 0x1000011)
 -> Resource User: Crash writer (ID: 0x1000012)
 -> Resource User: RO Notify Timers (ID: 0x1000013)
 -> Resource User: RMI RM Notify Watched Policy (ID: 0x1000014)
 -> Resource User: EnvMon (ID: 0x1000015)
 -> Resource User: OIR Handler (ID: 0x1000016)
 -> Resource User: IPC Dynamic Cache (ID: 0x1000017)
 -> Resource User: IPC Zone Manager (ID: 0x1000018)
 -> Resource User: IPC Periodic Timer (ID: 0x1000019)
 -> Resource User: IPC Managed Timer (ID: 0x100001A)
 -> Resource User: IPC Deferred Port Closure (ID: 0x100001B)
 -> Resource User: IPC Seat Manager (ID: 0x100001C)
 -> Resource User: IPC Session Service (ID: 0x100001D)
 -> Resource User: Compute SRP rates (ID: 0x100001E)
 -> Resource User: ARP Input (ID: 0x100001F)
 -> Resource User: DDR Timers (ID: 0x1000020)
 -> Resource User: Dialer event (ID: 0x1000021)
 -> Resource User: Entity MIB API (ID: 0x1000022)
 -> Resource User: SERIAL A'detect (ID: 0x1000023)
 -> Resource User: GraphIt (ID: 0x1000024)
 -> Resource User: HC Counter Timers (ID: 0x1000025)
 -> Resource User: Critical Bkgnd (ID: 0x1000026)
 -> Resource User: Net Background (ID: 0x1000027)
 -> Resource User: Logger (ID: 0x1000028)
```

```
.
.
.
Resource User Type: test_RUT141 (ID: 0x92)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT142 (ID: 0x93)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT143 (ID: 0x94)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT144 (ID: 0x95)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT145 (ID: 0x96)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT146 (ID: 0x97)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT147 (ID: 0x98)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT148 (ID: 0x99)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT149 (ID: 0x9A)
 -> Resource Owner: test_RO0 (ID: 0x7)
```

Step 13 **show resource user** {**all** | *resource-user-type*} [**brief** | **detailed**]
Use this command to display the relationship details between different ROs, for example:

Example:

```
Router# show resource user all
Resource User Type: iosprocess
Resource Grp: Init
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
27197780  8950144 18247636     6552
I/O memory
Allocated   Freed  Holding   Blocks
 7296000     9504  7286496      196
Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777224       14408       116     124206 100.40%  8.20%  1.70% Init
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
332      60       272      Init
Resource User: Init
Resource User: Scheduler
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
   77544        0    77544        2
Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        Scheduler
Resource User: Dead
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
 1780540      260  1780280      125
.
.
.
 Resource User: BGP Scanner
  Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
    9828     9828        0        0
  Resource Owner: cpu
```

```
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777406         660       659       1001  0.00%  0.00%  0.00% BGP Scanner
  Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        BGP Scanner
Resource User Type: test_process
Resource User Type: mem_rut
Resource User Type: cpu_rut
```

Troubleshooting Tips

To trace and troubleshoot the notification and registration activities for resources using the Embedded Resource Manager feature, use the following suggested techniques.

- Enable debugging of resource registration using the **debug resource policy registration**command in privileged EXEC mode.
- Enable debugging of resource manager notification using the **debug resource policy notification**command in privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **debug resource policy registration**
3. **debug resource policy notification** [**owner** *resource-owner-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug resource policy registration** **Example:** `Router# debug resource policy registration`	Enables debugging on resource policy registration.
Step 3	**debug resource policy notification** [**owner** *resource-owner-name*] **Example:** `Router# debug resource policy notification owner cpu`	Enables notification debugging on ROs.

Examples

Use the **debug resource policy registration**command to trace the resource manager registration information, for example:

```
Router# debug resource policy registration
Registrations debugging is on
When a Resource User is created
*Mar  3 09:35:58.304: resource_user_register: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0
name: usrr1
When a Resource User is deleted
*Mar  3 09:41:09.500: resource_user_unregister: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0
 name: usrr1
```

Use the **debug resource policy notification** [**owner** *resource-owner-name*] command to trace the resource policy notification information, for example:

```
Router# debug resource policy notification

Enabled notif. debugs on all owners
```

When a threshold is exceeded, you would see these messages:

```
*Mar  3 09:50:44.081: Owner: 'memory' initiated a notification:
*Mar  3 09:50:44.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory
 threshold
Pool: Processor Used: 42932864 Threshold :42932860
*Mar  3 09:50:46.081: Notification from Owner: 'memory' is dispatched for User: 'usrr1'
(ID: 0x10000B9)
*Mar  3 09:50:46.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory
 threshold
Pool: Processor Used: 42932864 Threshold :42932860
Router# no debug resource policy notification
Disabled notif. debugs on all owners
Router# debug resource policy notification owner cpu
Enabled notif. debugs on owner 'cpu'
Router#
no debug resource policy notification owner cpu

Disabled notif. debugs on owner 'cpu'
Router#
debug resource policy notification owner memory
Enabled notif. debugs on owner 'memory'
Router#
no debug resource policy notification owner memory
Disabled notif. debugs on owner 'memory'
Router#
debug resource policy notification owner Buffer
Enabled notif. debugs on owner 'Buffer'
Router#
no debug resource policy notification owner Buffer
Disabled notif. debugs on owner 'Buffer'
```

Configuration Examples for Embedded Resource Manager

Managing Resource Utilization by Defining Resource Policy Example

The following example shows how to configure a global resource policy with the policy name system-global-pc1:

```
configure terminal
resource policy
policy system-global-pc1 global
```

The following example shows how to configure a per user global resource policy with the policy name per-user-global-pc1 and the resource type as iosprocess:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
```

The following example shows how to configure a user local resource policy with the policy name user-local-pc1 and the resource type as iosprocess:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
```

Setting Expected Operating Ranges for Resource Owners Example

The following example shows how to configure various thresholds for buffer, CPU, and memory ROs.

Configuring System Global Thresholding Policy for Buffer RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
buffer public
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Buffer RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at

an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
buffer public
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Buffer RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 for public buffer with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
buffer public
critical rising 70 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for I/O Memory RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
memory io
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for I/O Memory RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
memory io
critical rising 90 interval 12 falling 20 interval 10 global
```

```
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for I/O Memory RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 for I/O memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
memory io
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Processor Memory RO

The following example shows how to configure a user system global policy with the policy name as system-global-pc1 for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Processor Memory RO

The following example shows how to configure a per user global policy with the policy name as user-global-pc1 and the resource type as iosprocess for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-global-pc1 type iosprocess
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring User Local Thresholding Policy for Processor Memory RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for processor memory with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70

percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
memory processor
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Interrupt CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for interrupt CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 global type iosprocess
```

```
system
cpu interrupt
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Process CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
cpu process
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Process CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
resource policy per-user-global-pc1 type iosprocess
system
cpu process
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Process CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for process CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 global type iosprocess
system
cpu process
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring System Global Thresholding Policy for Total CPU RO

The following example shows how to configure a global policy with the policy name as system-global-pc1 for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy system-global-pc1 global
system
cpu total
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Configuring Per User Global Thresholding Policy for Total CPU RO

The following example shows how to configure a per user global policy with the policy name as per-user-global-pc1 and the resource type as iosprocess for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy per-user-global-pc1 type iosprocess
system
cpu total
critical rising 90 interval 12 falling 20 interval 10 global
major rising 70 interval 12 falling 15 interval 10 global
minor rising 60 interval 12 falling 10 interval 10 global
```

Configuring User Local Thresholding Policy for Total CPU RO

The following example shows how to configure a user local policy with the policy name as user-local-pc1 and the resource type as iosprocess for total CPU with critical threshold values of 90 percent as rising at an interval of 12 seconds, 20 percent as falling at an interval of 10 seconds, major threshold values of 70 percent as rising at an interval of 12 seconds, 15 percent as falling at an interval of 10 seconds, and minor threshold values of 60 percent as rising at an interval of 12 seconds, 10 percent as falling at an interval of 10 seconds:

```
configure terminal
resource policy
policy user-local-pc1 type iosprocess
system
cpu total
critical rising 90 interval 12 falling 20 interval 10
major rising 70 interval 12 falling 15 interval 10
minor rising 60 interval 12 falling 10 interval 10
```

Applying a Policy Example

The following example shows how to apply a per user thresholding policy for the resource instance EXEC, resource user type iosprocess, and policy name policy-test1:

```
configure terminal
resource policy
```

```
policy policy-test1 type iosprocess
exit
user EXEC iosprocess policy-test1
```

The following example shows how to apply a global thresholding policy with the policy name global-global-test1:

```
configure terminal
resource policy
policy global-global-test1 global
exit
user global global-global-test1
```

The following example shows how to apply a group thresholding policy with the group name gr1 and resource type as iosprocess:

```
configure terminal
resource policy
policy group-test1
exit
user group gr1 type iosprocess
instance http
policy group-test1
```

Setting a System Global Thresholding Policy for I O Memory Example

The following example shows the configuration of a global memory thresholding policy for I/O memory. In this example, the policy is given the name “system-global-io”, and the threshold for critical I/O memory usage is defined as being usage of over 90 percent of the globally available I/O memory pool for 12 consecutive seconds.

The critical falling threshold is also defined in this example (less than 20 percent of the globally available I/O memory pool for 10 seconds or more); however, only the critical rising level will affect when the automatic deallocation procedure is triggered.

```
configure terminal
resource policy
policy system-global-io global
system
memory io
critical rising 90 interval 12 falling 20 interval 10
```

Additional References

The following sections provide references related to Embedded Resource Manager.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuration fundamentals commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Configuration Fundamentals Command Reference*

Related Topic	Document Title
Network management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Embedded Event Manager configuration tasks	Cisco IOS Embedded Event Manager
Memory Leak Detector	Memory Leak Dectector

Standards

Standards	Title
No new or modified standards are supported by this feature.	--

MIBs

MIBs	MIBs Link
CISCO-ERM-MIB.my	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

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Embedded Resource Manager

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Embedded Resource Manager

Feature Name	Releases	Feature Information
Embedded Resource Manager	12.3(14)T 12.2(33)SRB 12.2(33)SB Cisco IOS XE Release 3.9S	The Embedded Resource Manager (ERM) feature allows you to monitor internal system resource utilization for finite resources such as the buffer, memory, and CPU. ERM monitors resource utilization from the perspective of various subsystems within the Cisco IOS software such as resource owners (ROs) and resource users (RUs). ERM allows you to configure threshold values for system resources, leading to better insight into system scalability and improved system availability.

Feature Name	Releases	Feature Information
Embedded Resource Manager MIB	15.0(1)M 12.2(33)SRB 12.2(33)SB Cisco IOS XE Release 3.9S	The ERM MIB feature introduces MIB support for the Embedded Resource Manager (ERM) feature. The ERM feature tracks resource usage information for every registered resource owner and resource user. ERM ensures efficient usage of available resources.The ERM MIB feature allows you to monitor the usage of resources by gathering resource usage information using MIB objects. The network manager can use the information collected by the ERM MIB objects to ensure the optimal use of the resources. The following command was introduced by this feature: **snmp-server enable traps resource-policy.**
Packet Memory Reclamation	12.4(6)T 12.2(33)SRE Cisco IOS XE Release 3.9S	The Packet Memory Reclamation functionality utilizes the ERM infrastructure to cleanup and reclaim leaked Cisco IOS packet memory using the Memory Leak Detector process (sometimes referred to as the “Garbage Detection” or “GD” process).

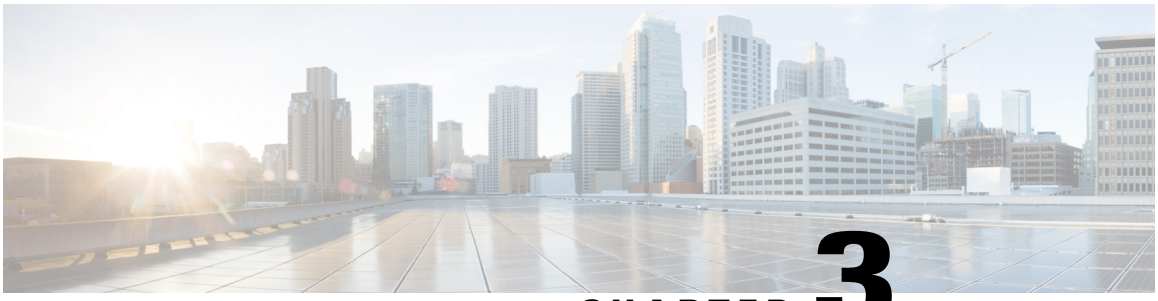
Glossary

CPUHOG --Each process is allocated a quantum of time, which is equivalent to 200 ms. If a process is running for more than 2 seconds, the process is hogging the CPU. This condition is called CPUHOG.

RM --resource usage monitors. Applications that wants to monitor resource utilization of resources by the resource users.

RO --resource owners. Provides resources to the resource users. For example, CPU, buffer, memory and so on.

RU --resource users. Applications or clients (like HTTP, SNMP, telnet, and so on) that use the resources and receive notifications to throttle when the current values exceed thresholds.



CHAPTER 3

Configuring Embedded Resource Manager-MIB

The Embedded Resource Manager (ERM)-MIB feature introduces MIB support for the ERM feature. The ERM feature tracks resource usage information for every registered resource owner and resource user. The ERM-MIB feature allows you to monitor the usage of resources by gathering resource usage information using MIB objects. The network manager can use the information collected by the ERM-MIB objects to ensure the optimal use of the resources.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for ERM-MIB

Simple Network Management Protocol (SNMP) must be enabled on the router before notifications (traps) can be configured or before SNMP GET operations can be performed.

Information About ERM-MIB

The ERM-MIB feature introduces network management support for ERM through the use of ERM-MIB table entries, MIB objects, and MIB trap notification objects that are defined in CISCO-ERM-MIB.my.

To use the ERM-MIB feature, you should understand the following concepts:

ERM Show MIB Objects

The ERM Show MIB objects are read-only objects. You can use these MIB objects to obtain information about resource owners, resource user type, resource users or groups, resource owner and resource user relationships, and resource monitors in the system.

The table below describes the ERM Show MIB objects.

Table 2: ERM Show MIB objects

ERM Show MIB Objects	Purpose
cermResOwnerTable	Obtains the details of all resource owners in the system.
cermResOwnerSubTypeTable	Obtains the details of the resource owner sub-types in the system.
cermResOwnerSubTypeThresholdTable	Obtains the details of the threshold value defined for each resource owner sub-type in the system.
cermResUserTypeTable	Obtains the details of the resource user types in the system.
cermResUserTable	Obtains the details of each resource user in the system.
cermResGroupTable	Obtains the details of each resource group in the system.
cermResGroupResUserTable	Obtains the details of resource users available in a specific resource group.
cermResOwnerResUserOrGroupTable	Obtains the details of all the resource owners, resource users, and group relationships defined in the system.
cermResOwnerResUserOrGroupThresholdTable	Obtains the details of the threshold value defined for each resource owner sub-type, resource user or resource group relationship in the system.
cermResUserTypeResOwnerTable	Obtains the details of resource owners present in a specific resource user type.

ERM Show MIB Objects	Purpose
cermResMonitorTable	Obtains the details of resource monitors in the system.
cermResMonitorResOwnerResUserTable	Obtains the details of resource owners, resource users, and resource owner and resource user relationships that are monitored by a resource monitor.
cermResMonitorPolicyTable	Obtains the details of resource policies that are monitored by a resource monitor.

Obtaining Information About Resource Owners

You can use cermResOwnerTable to obtain information about all resource owners in the system. The index entries for cermResOwnerTable are entPhysicalIndex, cermResOwnerSubEntityId, and cermResOwnerId.

The cermResOwnerTable defines the following MIB objects:

- cermResOwnerSubEntityId
- cermResOwnerId
- cermResOwnerName
- cermResOwnerMeasurementUnit
- cermResOwnerThresholdIsConfigurable
- cermResOwnerResUserCount
- cermResOwnerResGroupCount

Obtaining Sub-type Specific Information

You can use cermResOwnerSubTypeTable to obtain sub-type specific information. The cermResOwnerSubTypeTable is an extension of the cermResOwnerTable. The index entries for cermResOwnerSubTypeTable are entPhysicalIndex, cermResOwnerSubEntityId, cermResOwnerId, and cermResOwnerSubTypeId.

Each resource owner will have one or more entries in this table. For example, the CPU resource owner has three sub-types: process, interrupt, and total.

Some resource owners may not have any sub-types, such as the IPC resource owner. In such cases this table will contain a single entry with cermResOwnerSubTypeId as 0 and cermResOwnerSubTypeName as an empty string.

You can obtain all sub-type related information specified in this table by querying the corresponding resource owner.

The cermResOwnerSubTypeTable defines the following objects:

- cermResOwnerSubTypeId
- cermResOwnerSubTypeName

- cermResOwnerSubTypeUsagePct
- cermResOwnerSubTypeUsage
- cermResOwnerSubTypeMaxUsage
- cermResOwnerSubTypeGlobNotifSeverity

Obtaining Applied System Global Threshold Details

You can use cermResOwnerSubTypeThresholdTable to obtain applied threshold details for each resource owner sub-type. This object is an extension of the cermResOwnerSubTypeTable.

The index entries for cermResOwnerSubTypeThresholdTable are entPhysicalIndex, cermResOwnerSubEntityId, cermResOwnerId, cermResOwnerSubTypeId, and cermResOwnerSubTypeThreshSeverity. You can obtain all threshold details corresponding to a resource owner sub-type by querying the corresponding resource owner.

The cermResOwnerSubTypeThresholdTable defines the following objects:

- cermResOwnerSubTypeThreshSeverity
- cermResOwnerSubTypeRisingThresh
- cermResOwnerSubTypeRisingInterval
- cermResOwnerSubTypeFallingThresh
- cermResOwnerSubTypeFallingInterval

Obtaining Information About a Resource User Type

You can use cermResUserTypeTable to obtain information about a resource user type. Each resource user type in the system has an entry in cermResUserTypeTable. The index entries for this object are entPhysicalIndex, cermResUserTypeSubEntityId, and cermResUserTypeId.

The cermResUserTypeTable defines the following objects:

- cermResUserTypeSubEntityId
- cermResUserTypeId
- cermResUserTypeName
- cermResUserTypeResOwnerCount
- cermResUserTypeResUserCount
- cermResUserTypeResGroupCount

Obtaining Resource User-Specific Information

You can use cermResUserTable to obtain information about each resource user in the system. This object is an extension of cermResUserTypeTable. The index entries for cermResUserTable are entPhysicalIndex, cermResUserTypeSubEntityId, cermResUserTypeId, and ermResUserId.

The cermResUserTable defines the following objects:

- cermResUserId
- cermResUserName
- cermResUserPriority
- cermResUserResGroupId

Obtaining Information About Resource Groups

You can use cermResGroupTable to obtain information about every resource group available in the system. This object is an extension of cermResUserTypeTable. The index entries for cermResGroupTable are entPhysicalIndex, cermResUserTypeSubEntityId, cermResUserTypeId, and cermResGroupId.

The cermResGroupTable defines the following objects:

- cermResGroupId
- cermResGroupName
- cermResGroupUserInstanceCount

Obtaining Information About Resource Users in a Particular Resource Group

You can use cermResGroupResUserTable to obtain the list of resource users available in a particular resource group. This object is an extension of cermResGroupTable. The index entries for cermResGroupResUserTable are entPhysicalIndex, cermResUserTypeSubEntityId, cermResUserTypeId, cermResGroupId, and cermResGroupResUserId.

The cermResGroupResUserTable defines the following object:

- cermResGroupResUserId

Obtaining Information About Resource Owner and User Relationships

You can use cermResOwnerResUserOrGroupTable to obtain information about each resource owner- user relationship or resource owner-group relationship in the system. This object is an extension of cermResOwnerSubTypeTable.

The index entries for cermResOwnerResUserOrGroupTable are entPhysicalIndex, cermResOwnerSubEntityId, cermResOwnerId, cermResOwnerSubTypeId, cermResOwnerResUserTypeId, and cermResOwnerResUserOrGroupId.

This table can be used for the following tasks:

- To obtain the list of resource users registered for a specific resource owner.
- To obtain usage, max-usage, user local and per user global current notification levels for a given resource owner sub-type and resource user relation.

The cermResOwnerResUserOrGroupTable defines the following objects:

- cermResOwnerResUserTypeId
- cermResOwnerResUserOrGroupId

- cermResUserOrGroupFlag
- cermResUserOrGroupUsagePct
- cermResUserOrGroupUsage
- cermResUserOrGroupMaxUsage
- cermResUserOrGroupNotifSeverity
- cermResUserOrGroupGlobNotifSeverity

Obtaining Threshold Information About Each Resource Owner Sub-type and Resource User Relationship

You can use cermResOwnerResUserOrGroupThresholdTable to obtain threshold information about each resource owner sub-type and resource user relationship. This object is an extension of the cermResOwnerResUserOrGroupTable.

The index entries for cermResOwnerResUserOrGroupThresholdTable are entPhysicalIndex, cermResOwnerSubEntityId, cermResOwnerId, cermResOwnerSubTypeId, cermResOwnerResUserTypeId, cermResOwnerResUserOrGroupId, cermResUserOrGroupThreshIsUserGlob, and cermResUserOrGroupThreshSeverity.

The cermResOwnerResUserOrGroupThresholdTable defines the following objects:

- cermResUserOrGroupThreshIsUserGlob
- cermResUserOrGroupThreshSeverity
- cermResUserOrGroupThreshFlag
- cermResUserOrGroupRisingThresh
- cermResUserOrGroupRisingInterval
- cermResUserOrGroupFallingThresh
- cermResUserOrGroupFallingInterval

Obtaining Information About Resource Owners Present in a Resource User Type

You can use cermResUserTypeResOwnerTable to obtain the list of resource owners present in a resource user type. This object is an extension of the cermResUserTypeTable.

The index entries for cermResUserTypeResOwnerTable are entPhysicalIndex, cermResUserTypeSubEntityId, cermResUserTypeId, and cermResUserTypeResOwnerId.

The cermResUserTypeResOwnerTable defines the following objects:

- cermResUserTypeResOwnerId

Obtaining Information About Resource Monitors

You can use cermResMonitorTable to obtain the list of resource monitors in the system. The index entries for this object are entPhysicalIndex, cermResMonitorSubEntityId, and cermResMonitorId.

The cermResMonitorTable defines the following objects:

- cermResMonitorSubEntityId
- cermResMonitorId
- cermResMonitorName

Obtaining Resource Information About Resource Owner and User Relationships that are Monitored

You can use cermResMonitorResOwnerResUserTable to obtain resource-related information that is tracked by a resource monitor. This object is an extension of cermResMonitorTable.

The index entries for cermResMonitorResOwnerResUserTable are entPhysicalIndex, cermResMonitorSubEntityId, cermResMonitorId, cermResMonitorResOwnerId, cermResMonitorResUserTypeId, and cermResMonitorResUserId.

The cermResMonitorResOwnerResUserTable defines the following objects:

- cermResMonitorResOwnerId
- cermResMonitorResUserTypeId
- cermResMonitorResUserId
- cermResMonitorResPolicyName

Obtaining Information About Resource Policies that are Monitored by a Resource Monitor

You can use cermResMonitorPolicyTable to obtain the list of resource policies that are tracked by a resource monitor. This object is an extension of the cermResMonitorTable. The index entries for cermResMonitorPolicyTable are entPhysicalIndex, cermResMonitorSubEntityId, cermResMonitorId, and cermResMonitorPolicyName.

The cermResMonitorPolicyTable defines the following object:

- cermResMonitorPolicyName

ERM Configuration MIB Objects

You can use the ERM Configuration MIB objects to perform the following tasks:

The table below describes the ERM Configuration MIB objects.

Table 3: ERM Configuration MIB Objects

ERM Configuration MIB Objects	Purpose
cermScalarsGlobalPolicyName (scalar object)	Identifies and indicates the global resource policy applied in the system.
cermConfigPolicyTable	Creates, modifies, or deletes a resource policy.

ERM Configuration MIB Objects	Purpose
cermConfigPolicyResOwnerThreshTable	Configures threshold values and intervals for resource owner sub-types.
cermConfigResGroupTable	Creates or deletes a resource group.
cermConfigResGroupUserTable	Creates or deletes a user instance in a resource group.
cermConfigPolicyApplyTable	Applies an existing resource policy to a resource user or group.

Verifying Whether a Global Resource Policy Is Applied in the System

You can use the scalar object cermScalarsGlobalPolicyName to identify and indicate if a global resource policy is applied in the system. If no global resource policy is applied in the system, this object will contain an empty string. This object has read-write access permission. Setting this scalar object to an existing global resource policy name will result in applying the global resource policy to the system.

Creating Modifying or Deleting a Resource Policy

You can use cermConfigPolicyTable to create, modify, or delete a resource policy. The index entry for this object is cermPolicyName.

The cermConfigPolicyTable defines the following objects:

- cermPolicyName
- cermPolicyIsGlobal
- cermPolicyUserTypeName
- cermPolicyLoggingEnabled
- cermPolicySnmpNotifEnabled
- cermPolicyStorageType
- cermPolicyRowStatus

Configuring Threshold Values and Intervals for Resource Owner Sub-types in a Resource Policy

You can use cermConfigPolicyResOwnerThreshTable to configure rising or falling threshold values and rising or falling intervals for resource owner sub-types in a resource policy. This object is an extension of the cermConfigPolicyTable.

The index entries for cermConfigPolicyResOwnerThreshTable are cermPolicyName, cermPolicyPhysicalIndex, cermConfigPolicyResOwnerSubEntityId, cermConfigPolicyResOwnerId, cermConfigPolicyResOwnerSubTypeId, ermConfigPolicyIsUserGlobal, and cermConfigPolicyThresholdLevel.

The cermConfigPolicyResOwnerThreshTable defines the following objects:

- cermPolicyPhysicalIndex
- cermConfigPolicyResOwnerSubEntityId
- cermPolicyResOwnerId
- cermPolicyResOwnerSubTypeId
- cermPolicyIsUserGlobal
- cermPolicyThresholdLevel
- cermPolicyRisingThreshold
- cermPolicyRisingInterval
- cermPolicyFallingThreshold
- cermPolicyFallingInterval
- cermPolicyResOwnerThreshStorageType
- cermPolicyResOwnerRowStatus

Creating or Deleting a Resource Group

You can use cermConfigResGroupTable to create or delete a resource group in the system. The index entry for this object is cermConfigResGroupName.

The cermConfigResGroupTable defines the following objects:

- cermConfigResGroupName
- cermConfigResGroupUserTypeName
- cermConfigResGroupStorageType
- cermConfigResGroupRowStatus

Creating or Deleting a User Instance in a Resource Group

You can use cermConfigResGroupUserTable to create or delete a user instance in a given resource group. This object is an extension of the cermConfigResGroupTable.

The index entries for cermConfigResGroupUserTable are cermConfigResGroupName and cermConfigResGroupUserName.

The cermConfigResGroupUserTable defines the following objects:

- cermConfigResGroupUserName
- cermConfigResGroupUserStorageType
- cermConfigResGroupUserRowStatus

Applying an Existing Resource Policy to a Resource User or Group

You can use cermConfigPolicyApplyTable to apply an existing resource policy to a resource user or resource group. The index entries for this object are cermPolicyApplyUserOrGroupName and cermPolicyApplyUserOrGroupFlag.

The cermConfigPolicyApplyTable defines the following objects:

- cermPolicyApplyUserOrGroupName
- cermPolicyApplyUserOrGroupFlag
- cermPolicyApplyPolicyName
- cermPolicyApplyStorageType
- cermPolicyApplyRowStatus

ERM Notification MIB Objects

You can configure ERM Notification MIB objects to receive global or user-specific notification on policy violation. There are three types of ERM Notification MIB objects.

The table below describes the ERM Notification MIB objects.

Table 4: ERM Notification MIB Objects

ERM Notification MIB Objects	Purpose
cermNotifsEnabled	Enables ERM notifications.
ciscoErmGlobalPolicyViolation	Specifies the type of notification received on global policy violation.
ciscoErmLocalPolicyViolation	Specifies the type of user-specific notification received on local policy violation.

Controlling the Generation of Traps for ERM Policy Violation Notifications

You can use cermNotifsEnabled to determine if the generation of traps for ERM policy violation notifications is allowed.

When this object is set to true, it allows generation of traps for the ERM policy violation related notifications ciscoErmGlobalPolicyViolation and ciscoErmLocalPolicyViolation.

Receiving a Global Notification on Policy Violation

You can use ciscoErmGlobPolicyViolation to receive global notification on policy violation.

The notification object ciscoErmGlobPolicyViolation defines the following objects:

- cermResOwnerName
- cermResOwnerSubTypeName
- cermNotifsThresholdSeverity
- cermNotifsThresholdValue
- cermNotifsDirection
- cermNotifsPolicyName

Receiving a User-Specific Notification on Policy Violation

You can use ciscoErmUserPolicyViolation to receive a user-specific notification on policy violation.

The notification object ciscoErmUserPolicyViolation contains the following objects:

- cermResOwnerName
- cermResOwnerSubTypeName
- cermResUserTypeName
- cermResUserName
- cermResUserOrGroupThreshFlag
- cermNotifsThresholdIsUserGlob
- cermNotifsThresholdSeverity
- cermNotifsThresholdValue
- cermNotifsDirection
- cermNotifsPolicyName

How to Configure ERM-MIB

Enabling ERM-MIB Notification Traps

You can enable ERM-MIB notification traps, which are generated when resource usage exceeds the threshold value. The ERM-MIB notification traps will be sent to the host that is configured to receive traps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps resource-policy**
4. **end**

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.
Step 3	**snmp-server enable traps resource-policy** **Example:** `Router(config)# snmp-server enable traps resource-policy`	Enables CISCO-ERM-MIB notifications.
Step 4	**end** **Example:** `Router(config)# end`	Returns the router to privileged EXEC mode.

Configuring the Router to Send SNMP Notification Traps for ERM to a Host

Perform this task to enable the router to send SNMP notifications traps defined in ERM-MIB to a host.

Before You Begin

- SNMP must be enabled on your network.
- Create an SNMP server community to receive information on MIB objects and traps using the **snmp-server community** command.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]
3. **configure terminal**
4. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [*options*] **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 209.165.201.30 traps version 2c priv mycommunitystring isis`	Specifies the recipient (target host) for ERM SNMP notification operations.
Step 5	**end** **Example:** `Router(config)# end`	Returns the router to privileged EXEC mode.

Configuration Examples for ERM-MIB

Configuring the Router to Send SNMP Notifications for ERM to a Host Example

The following example shows how to configure the router to send SNMP notifications for ERM to a host:

```
Router# configure terminal
Router(config)# snmp-server community public rw
```

```
Router(config)# snmp-server enable traps resource-policy
Router(config)# snmp-server host 209.165.201.30 version 2c public
Router(config)# end
```

Additional References

The following sections provide references related to the ERM-MIB feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Embedded Resource Manager	*Embedded Resource Manager*
Network Management commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
• CISCO-ERM-MIB.my	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

RFCs

RFC	Title
RFC 1902	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ERM-MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for ERM-MIB

Feature Name	Releases	Feature Information
Embedded Resource Manager (ERM)-MIB	12.2(33)SB 12.2(33)SRB 12.4(15)T Cisco IOS XE Release 3.9S	The ERM-MIB feature introduces MIB support for the Embedded Resource Manager (ERM) feature. The ERM-MIB feature allows you to monitor the usage of resources by gathering resource usage information using MIB objects. The network manager can use the information collected by the ERM-MIB objects to ensure the optimal use of the resources. The following commands were introduced or modified by this feature: **snmp-server enable traps resource-policy**