THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2011 Cisco Systems, Inc. All rights reserved.
CONTENTS

Embedded Resource Manager (ERM) 1
  Finding Feature Information 1
  Prerequisites for Embedded Resource Manager 1
  Restrictions for Embedded Resource Manager 2
  Information About Embedded Resource Manager 2
    Benefits of the Embedded Resource Manager 2
    Resource Accounting and Thresholds Tracking in ERM 3
    System Resources Monitored by the Embedded Resource Manager 3
      CPU Resource Owner 3
        Loadometer Process 4
        Scheduler 4
        Snapshot Management Using Event Trace 4
        Automatic CPUHOG Profiling 4
      Memory Resource Owner 4
        Memory Usage History 4
        Memory Accounting 4
        Interface Wedging and Packet Memory Leaks 5
        Memory Resource Reclamation for Interfaces 5
        Memory Leak Reclamation 6
        I O Memory 6
      Buffer Resource Owner 6
        Automatic Buffer Tuning 6
        Buffer Leak Detection 7
        Buffer Accounting 7
        Buffer Usage Thresholding 7
  Resource Policy Templates 8
How to Configure Embedded Resource Manager 8
  Managing Resource Utilization by Defining Resource Policy 8
  Setting Expected Operating Ranges for Buffer Resources 9
Obtaining Information About Resource Monitors 66
Obtaining Resource Information About Resource Owner and User Relationships that are Monitored 66
Obtaining Information About Resource Policies that are Monitored by a Resource Monitor 67

ERM Configuration MIB Objects 67
Verifying Whether a Global Resource Policy Is Applied in the System 68
Creating Modifying or Deleting a Resource Policy 68
Configuring Threshold Values and Intervals for Resource Owner Sub-types in a Resource Policy 68
Creating or Deleting a Resource Group 69
Creating or Deleting a User Instance in a Resource Group 69
Applying an Existing Resource Policy to a Resource User or Group 69

ERM Notification MIB Objects 69
Controlling the Generation of Traps for ERM Policy Violation Notifications 70
Receiving a Global Notification on Policy Violation 70
Receiving a User-Specific Notification on Policy Violation 70

How to Configure ERM-MIB 71
Enabling ERM-MIB Notification Traps 71
Configuring the Router to Send SNMP Notification Traps for ERM to a Host 72

Configuration Examples for ERM-MIB 73
Configuring the Router to Send SNMP Notifications for ERM to a Host Example 73

Additional References 73
Feature Information for ERM-MIB 74
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, page 1
- Prerequisites for Embedded Resource Manager, page 1
- Restrictions for Embedded Resource Manager, page 2
- Information About Embedded Resource Manager, page 2
- How to Configure Embedded Resource Manager, page 8
- Configuration Examples for Embedded Resource Manager, page 49
- Additional References, page 55
- Feature Information for Embedded Resource Manager, page 57
- Glossary, page 58

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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, page 2
- Resource Accounting and Thresholds Tracking in ERM, page 3
- System Resources Monitored by the Embedded Resource Manager, page 3
- Resource Policy Templates, page 8

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.

![ERM Architecture Diagram](image)

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 RU’s 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, page 3
• Memory Resource Owner, page 4
• Buffer Resource Owner, page 6

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, page 4
• Scheduler, page 4
• Snapshot Management Using Event Trace, page 4
• Automatic CPUHOG Profiling, page 4
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, page 4
- Memory Accounting, page 4
- Interface Wedging and Packet Memory Leaks, page 5
- Memory Resource Reclamation for Interfaces, page 5
- Memory Leak Reclamation, page 6
- I O Memory, page 6

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 Detector feature 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, page 6
• Buffer Leak Detection, page 7
• Buffer Accounting, page 7
• Buffer Usage Thresholding, page 7

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, page 8
- Setting Expected Operating Ranges for Buffer Resources, page 9
- Setting Expected Operating Ranges for CPU Resources, page 12
- Setting Expected Operating Ranges for Memory Resources, page 19
- Enabling Automatic Tuning of Buffers, page 25
- Managing Memory Usage History, page 25
- Configuring a CPU Process to Be Included in the Extended Load Monitor Report, page 26
- Managing Extended CPU Load Monitoring, page 27
- Managing Automatic CPUHOG Profiling, page 28
- Applying a Policy to Resource Users, page 28
- Setting a Critical Rising Threshold for Global I O Memory, page 30
- Verifying ERM Operations, page 33
- Troubleshooting Tips, page 47

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |

  **Example:**

  Router> enable

| **Step 2** configure terminal | Enters global configuration mode. |

  **Example:**

  Router# configure terminal

| **Step 3** resource policy | Enters ERM configuration mode. |

  **Example:**

  Router(config)# resource policy

| **Step 4** policy policy-name [global | type resource-user-type] | 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. |

  **Example:**

  Router(config-erm)# policy policy1  
  type iosprocess

---

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3 resource policy</td>
<td>Enters ERM configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# resource policy</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th><strong>policy policy-name [global] [type resource-user-type]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-erm)# policy policy1 type iosprocess</td>
</tr>
</tbody>
</table>

**Purpose**: 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.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Do one of the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>system</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>slot slot-number</strong></td>
</tr>
</tbody>
</table>

**Example**: Router(config-erm-policy)# system

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

**Example**: Router(config-erm-policy)# system

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

<table>
<thead>
<tr>
<th>Step 6</th>
<th><strong>buffer public</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-policy-node)# buffer public</td>
</tr>
</tbody>
</table>

**Purpose**: Enters buffer owner configuration mode.

- Allows you to set the rising and falling values for the critical, major, and minor thresholds.
### Command or Action

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do one of the following:</td>
<td>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.</td>
</tr>
<tr>
<td>• <strong>critical rising</strong> rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] [global]</td>
<td><strong>Note</strong> If you had configured a global policy in Step 4, you do not need to give the <strong>global</strong> 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 <strong>global</strong> keyword) in Step 4, enter the <strong>global</strong> keyword in Step 7 if you want to configure a per user global threshold.</td>
</tr>
<tr>
<td>• <strong>major rising</strong> rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] [global]</td>
<td></td>
</tr>
<tr>
<td>• <strong>minor rising</strong> rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] [global]</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Router(config-owner-buffer)# critical rising 40 falling 20 interval 10 global

**Example:**

Router(config-owner-buffer)# major rising 30 falling 15 interval 10 global

**Example:**

Router(config-owner-buffer)# minor rising 20 falling 10 interval 10 global

### Step 8

**exit**

Exits buffer owner configuration mode.

**Example:**

Router(config-owner-buffer)# exit

---

**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
   • resource
   • 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. exit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>enable</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

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

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

<p>| <strong>Step 4</strong> | Configures a resource policy and enters ERM policy configuration mode. |
| <strong>policy policy-name [global | type resource-user-type]</strong> | |
| <strong>Example:</strong> | • The policy-name argument identifies the name of the resource policy. |
| Router(config-erm)# policy policy1 type iosprocess | • 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. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> Do one of the following:</td>
<td></td>
</tr>
<tr>
<td>• system</td>
<td>Enters policy node configuration mode with the <code>system</code> command.</td>
</tr>
<tr>
<td>• slot slot-number</td>
<td>Enters ERM slot configuration mode with the <code>slot slot-number</code> command. This command is available only in distributed platforms like the RSP.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-erm-policy)# system

**Example:**

**Example:**

Router(config-erm-policy)# slot 1

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> cpu interrupt</td>
<td>(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-policy-node)# cpu interrupt
### Command or Action

**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:**

Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global

**Example:**

Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global

### Purpose

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.

**Step 8** exit

**Example:**

Router(config-owner-cpu)# exit

Exits the CPU owner configuration mode.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> cpu process</td>
<td><em>(Optional) Enters CPU owner configuration mode.</em> Allows you to set the rising and falling values for the critical, major, and minor thresholds.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-policy-node)# cpu process</td>
</tr>
<tr>
<td><strong>Step 10</strong> Do one of the following:</td>
<td>Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of process CPU utilization. <strong>Note</strong> If you had configured a global policy in Step 4, you do not need to give the <code>global</code> 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 <code>global</code> keyword) in Step 4, enter the <code>global</code> 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.</td>
</tr>
<tr>
<td>• <strong>critical rising rising-threshold-value [interval [interval-value]] [global]</strong></td>
<td></td>
</tr>
<tr>
<td>• <strong>major rising rising-threshold-value [interval [interval-value]] [global]</strong></td>
<td></td>
</tr>
<tr>
<td>• <strong>minor rising rising-threshold-value [interval [interval-value]] [global]</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 11 exit</strong></td>
<td>Exits the CPU owner configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-owner-cpu)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12 cpu total</strong></td>
<td>(Optional) Enters CPU owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-policy-node)# cpu total</td>
<td></td>
</tr>
</tbody>
</table>
### Setting Expected Operating Ranges for Memory Resources

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 13</strong> Do one of the following:</td>
<td>Allows you to set the rising and falling threshold values for critical, major, and minor levels of percentages of total CPU utilization.</td>
</tr>
<tr>
<td>• critical rising rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] global</td>
<td><strong>Note</strong> If you had configured a global policy in Step 4, you do not need to give the <code>global</code> 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 <code>global</code> keyword) in Step 4, enter the <code>global</code> keyword in Step 13 if you want to configure a per user global threshold.</td>
</tr>
<tr>
<td>• major rising rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] global</td>
<td>For total CPU utilization, you can configure either global thresholds or per user global thresholds. Hence, you must enter the <code>global</code> keyword either in Step 4 or in Step 13.</td>
</tr>
<tr>
<td>• minor rising rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] global</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-owner-cpu)# critical rising 40 falling 20 interval 10 global</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-owner-cpu)# major rising 30 falling 15 interval 10 global</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-owner-cpu)# minor rising 20 falling 10 interval 10 global</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> <strong>exit</strong></td>
<td>Exits CPU owner configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-owner-cpu)# exit</td>
<td></td>
</tr>
</tbody>
</table>
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 Detector 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**
# Embedded Resource Manager (ERM)

## How to Configure Embedded Resource Manager

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> resource policy</td>
<td>Enters ERM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# resource policy</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> policy policy-name [global</td>
<td>type resource-user-type]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-erm)# policy policy1 type iosprocess</td>
<td></td>
</tr>
</tbody>
</table>

- 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:**

```plaintext
Router(config-erm-policy)# system
```

**Example:**

```plaintext
Router(config-erm-policy)# slot 1
```

---

### Step 6

**memory io**

(Optional) Enters memory owner configuration mode. Allows you to set the rising and falling values for the critical, major, and minor thresholds.

**Example:**

```plaintext
Router(config-policy-node)# memory io
```
### Command or Action

**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:**

Router(config-owner-memory)# major rising 30 falling 15 interval 10 global

**Example:**

Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global

### Purpose

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.

### Step 8

**exit**

Exits memory owner configuration mode.

**Example:**

Router(config-owner-memory)# exit
### Command or Action

**Step 9** memory processor

**Example:**

Router(config-policy-node)# memory processor

**Purpose**

(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:

1. **critical rising** rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] [global]
2. **major rising** rising-threshold-value [interval interval-value] [falling falling-threshold-value [interval interval-value]] [global]
3. **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:**

Router(config-owner-memory)# major rising 30 falling 15 interval 10 global

**Example:**

Router(config-owner-memory)# minor rising 20 falling 10 interval 10 global

**Purpose**

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.
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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> buffer tune automatic</td>
<td>Enables automatic tuning of buffers.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# buffer tune automatic</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> memory statistics history table number-of-hours</td>
<td>Changes the time (number of hours) for which the memory log is maintained.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# memory statistics history table 48</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> process cpu extended history history-size</td>
<td>Enables you to change the history size of the extended collection report.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# process cpu extended history 24</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 autoprobe hog

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> processes cpu autoprobe hog</td>
<td>Enables automatic profiling of CPUHOG processes.</td>
</tr>
<tr>
<td>Example:</td>
<td>This function is enabled by default.</td>
</tr>
<tr>
<td>Router(config)# processes cpu autoprobe hog</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `enable` | Enables privileged EXEC mode.  
* Enter your password if prompted. |
| **Example:** | Router> enable |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:** | Router# configure terminal |
| **Step 3** `resource policy` | Enters ERM configuration mode. |
| **Example:** | Router(config)# resource policy |
| **Step 4** `policy policy-name [global | type resource-user-type]` | 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. |
| **Example:** | Router(config-erm)# policy policy1 type iosprocess |
| **Step 5** `exit` | Exits ERM policy configuration mode. |
| **Example:** | Router(config-erm)# exit |
### Command or Action

**Step 6**  
**user** {  
  *resource-instance-name*  
  *resource-user-type*  
  *resource-policy-name*  
  |  
  *global*  
  *global-policy-name*  
  |  
  *group*  
  *resource-group-name*  
  type  
  *resource-user-type*}  

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.

**Example:**  
Router(config-erm)# user group lowPrioUsers type iosprocess

### Example:

**Step 7**  
**instance**  
*instance-name*  

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.

**Example:**  
Router(config-res-group)# instance http

### Example:

**Step 8**  
**policy**  
*policy-name*  

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.

**Example:**  
Router(config-res-group)# policy group-policy1

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
   - Enter your password if prompted. |
| Example: | Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example: | Router# configure terminal |
| **Step 3** resource policy | Enters ERM configuration mode. |
| Example: | Router(config)# resource policy |
| **Step 4** policy policy-name [global | type resource-user-type] | 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. |
| Example: | Router(config-erm)# policy policy1 type iosprocess |
### Command or Action | Purpose
--- | ---
**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 RSP.

**Example:**

Router(config-erm-policy)# system

**Example:**

Router(config-erm-policy)# slot 1

**Step 6** memory io | (Optional) Enters memory owner configuration mode.
- Allows you to set the rising and falling values for the critical, major, and minor thresholds.

**Example:**

Router(config-policy-node)# memory io
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 {detail | back time | clock time | from-boot {seconds | detail | latest [detail]} | handle handle-number]
7. show processes cpu autoprobe 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
6488F85C 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 74 0 0 10 None None EEM ED Sy
6489252C E001204 Small 60 0 0 0 None None EEM ED Sy
64892C24 E001484 Small 74 0 0 10 None None EEM ED Sy
64892FA0 E0015C4 Small 74 0 0 0 None None EEM ED Sy
64893A14 E001984 Small 74 0 0 0 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
6517D52C E176D44 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
64895C8C E003004 Middl 173 0 0 10 None None EEM ED Sy
64896068 E003344 Middl 176 0 0 10 None None EEM ED Sy
64896E5E E004044 Middl 109 0 0 10 None None EEM ED Sy
64897C48 E004D44 Middl 194 0 0 10 None None EEM ED Sy
```
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
permanent: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
permanent: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: 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
Allocating PC Summary for allocated blocks in pool: Processor

<table>
<thead>
<tr>
<th>PC</th>
<th>Total</th>
<th>Count</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x6189A438</td>
<td>318520</td>
<td>1</td>
<td>RTPSPI</td>
</tr>
<tr>
<td>0x6205711C</td>
<td>237024</td>
<td>2</td>
<td>CCH323_CT</td>
</tr>
<tr>
<td>0x6080BE38</td>
<td>98416</td>
<td>2</td>
<td>Exec</td>
</tr>
<tr>
<td>0x606AD988</td>
<td>80256</td>
<td>1</td>
<td>Init</td>
</tr>
<tr>
<td>0x618F68A8</td>
<td>73784</td>
<td>1</td>
<td>CCSIP_UDP_SOCKET</td>
</tr>
<tr>
<td>0x61854A04</td>
<td>67640</td>
<td>1</td>
<td>QOS_MODULE_MAIN</td>
</tr>
<tr>
<td>0x606488C8</td>
<td>65592</td>
<td>1</td>
<td>CEF: Adjacency chunk</td>
</tr>
</tbody>
</table>
0x60635620 65592 1 CEF: 16 path chunk pool
0x615ECE58 65592 1 XTagATM VC chunk
0x6165ACF8 65592 1 eddri_self_event
0x608DE168 65592 1MallocLite
0x60857920 51020 11 Normal
0x6203BF88 42480 4 IPv6 CEF fib tables
0x60D7C7F14 32824 1 PPP Context Chunks

I/O memory
Free memory size : 14700024 Number of free blocks: 52
I/O memory
Free memory size : 14700024 Number of free blocks: 52

 Allocator PC Summary for allocated blocks in pool: I/O
<table>
<thead>
<tr>
<th>PC</th>
<th>Total</th>
<th>Count</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x60857934</td>
<td>3936000</td>
<td>60</td>
<td>FastEthernet0/</td>
</tr>
<tr>
<td>0x60857898</td>
<td>524800</td>
<td>8</td>
<td>FastEthernet0/0</td>
</tr>
<tr>
<td>0x601263CC</td>
<td>29120</td>
<td>7</td>
<td>Init</td>
</tr>
<tr>
<td>0x6082DB28</td>
<td>9048</td>
<td>23</td>
<td><em>Packet Data</em></td>
</tr>
<tr>
<td>0x60126344</td>
<td>8448</td>
<td>4</td>
<td>I/O</td>
</tr>
</tbody>
</table>

 Allocator PC Summary for free blocks in pool: I/O
<table>
<thead>
<tr>
<th>PC</th>
<th>Total</th>
<th>Count</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x608C5730</td>
<td>29391444</td>
<td>1</td>
<td>(coalesced)</td>
</tr>
<tr>
<td>0x608FC1F4</td>
<td>5376</td>
<td>28</td>
<td>(fragment)</td>
</tr>
<tr>
<td>0x6082DB28</td>
<td>4288</td>
<td>14</td>
<td>(fragment)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Bytes</th>
<th>Prev</th>
<th>Next</th>
<th>Ref</th>
<th>PrevF</th>
<th>NextF</th>
<th>Alloc PC</th>
<th>what</th>
</tr>
</thead>
<tbody>
<tr>
<td>644AAB70</td>
<td>00000001032</td>
<td>644AAB20</td>
<td>644AAFAF2</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>60450F8</td>
<td>Index Table Block</td>
</tr>
<tr>
<td>644AAFAC</td>
<td>0000000028</td>
<td>644AAB70</td>
<td>644AFFC</td>
<td>000</td>
<td>6448CB5C</td>
<td>607B2ADC</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>644AFFC</td>
<td>0000000076</td>
<td>644AAFAC</td>
<td>644AB07C</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>60818DE0</td>
<td>Init</td>
</tr>
<tr>
<td>6448CBDC</td>
<td>0000000028</td>
<td>6448CABD</td>
<td>6448C5BC</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>607F8380</td>
<td>Cond Debug definition</td>
</tr>
<tr>
<td>6448CBAC</td>
<td>0000000028</td>
<td>6448C5BC</td>
<td>6448C5BC</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>607F8380</td>
<td>Cond Debug definition</td>
</tr>
<tr>
<td>6489EF8C</td>
<td>00000000408</td>
<td>6489D8C</td>
<td>6489F158</td>
<td>001</td>
<td>6448C5BC</td>
<td>607B2ADC</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>6489F158</td>
<td>0000000064</td>
<td>6489EFC</td>
<td>6489F1C</td>
<td>000</td>
<td>6448C5BC</td>
<td>607B2ADC</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>6489F1CC</td>
<td>00000005004</td>
<td>6489F158</td>
<td>648AB5BC</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>60857920</td>
<td>Normal</td>
</tr>
<tr>
<td>6448CAEC</td>
<td>0000000028</td>
<td>6448C9AC</td>
<td>6448C5BC</td>
<td>001</td>
<td>6448F158</td>
<td>644949C8</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>6448C5BC</td>
<td>0000000028</td>
<td>6448C5BC</td>
<td>6448C5BC</td>
<td>001</td>
<td>6448F158</td>
<td>644949C8</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>64494978</td>
<td>0000000028</td>
<td>6449498</td>
<td>6449498</td>
<td>001</td>
<td>644949C8</td>
<td>607D72FC</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>644949C8</td>
<td>0000000028</td>
<td>6449498</td>
<td>6449498</td>
<td>001</td>
<td>644949C8</td>
<td>607D72FC</td>
<td>NameDB String</td>
<td></td>
</tr>
<tr>
<td>6449F278</td>
<td>0000000076</td>
<td>6449F278</td>
<td>6449F288</td>
<td>001</td>
<td>644949C8</td>
<td>60818DE0</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>654F278</td>
<td>0000000076</td>
<td>654F278</td>
<td>654F288</td>
<td>001</td>
<td>644949C8</td>
<td>60818DE0</td>
<td>Init</td>
<td></td>
</tr>
</tbody>
</table>

I/O memory

Free memory size : 14700024 Number of free blocks: 52

<table>
<thead>
<tr>
<th>Address</th>
<th>Bytes</th>
<th>Prev</th>
<th>Next</th>
<th>Ref</th>
<th>PrevF</th>
<th>NextF</th>
<th>Alloc PC</th>
<th>what</th>
</tr>
</thead>
<tbody>
<tr>
<td>0E00000000</td>
<td>0000000056</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0E17EF4C</td>
<td>60000000</td>
<td>(fragment)</td>
<td></td>
</tr>
<tr>
<td>0E00000000</td>
<td>0000000268</td>
<td>0E000000</td>
<td>0E0001AC</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>6082DB28</td>
<td><em>Packet Data</em></td>
</tr>
<tr>
<td>0E17E6CC</td>
<td>0000000268</td>
<td>0E17E6CC</td>
<td>0E17E6FC</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>6082DB28</td>
<td><em>Packet Data</em></td>
</tr>
<tr>
<td>0E17E6FC</td>
<td>0000000076</td>
<td>0E17E6CC</td>
<td>0E17E6CC</td>
<td>000</td>
<td>E00000000</td>
<td>E029F4C</td>
<td>6082DB28</td>
<td>(fragment)</td>
</tr>
<tr>
<td>0E176FCC</td>
<td>0000002060</td>
<td>0E176F4C</td>
<td>0E17780C</td>
<td>001</td>
<td>--------</td>
<td>--------</td>
<td>60126344</td>
<td>Init</td>
</tr>
<tr>
<td>0E2096DC</td>
<td>0000002068</td>
<td>0E2096DC</td>
<td>0E2096FC</td>
<td>001</td>
<td>0E2096FC</td>
<td>60126344</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>0E2096FC</td>
<td>0000000076</td>
<td>0E2096EC</td>
<td>0E2096EC</td>
<td>001</td>
<td>E176F4C</td>
<td>E29FC4C</td>
<td>6082DB28</td>
<td>(fragment)</td>
</tr>
<tr>
<td>0E209F4C</td>
<td>0000002060</td>
<td>0E209F4C</td>
<td>0E20A80C</td>
<td>001</td>
<td>0E209F4C</td>
<td>60126344</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>0E209F4C</td>
<td>0000002068</td>
<td>0E209F4C</td>
<td>0E209F4C</td>
<td>001</td>
<td>0E209F4C</td>
<td>60126344</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>0E32FDEC</td>
<td>0000002068</td>
<td>0E32FDEC</td>
<td>0E32F4C</td>
<td>001</td>
<td>0E32F4C</td>
<td>60126344</td>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>0E32FDEC</td>
<td>0000002068</td>
<td>0E32FDEC</td>
<td>0E32F4C</td>
<td>001</td>
<td>0E32F4C</td>
<td>60126344</td>
<td>Init</td>
<td></td>
</tr>
</tbody>
</table>
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 607C2D20 Init
200B42B4 0000000028 200B4268 200B4300 001 -------- -------- 62366C80 Init
200B4300 0000000028 200B42B4 200B434C 000 645A8194 6490F7E8 60976574 AAA Event Data
6490F79C 0000000028 6490F748 6490F7E8 000 200B4300 6491120C 606DD8D8 Init
6490F7E8 0000000028 6490F79C 6490F834 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.
20000000 0000000828 5C3AEB24 2000036C 001 -------- -------- 60734010 *Packet Header*
6500BF94 0000000828 6500BC28 6500C300 001 -------- -------- 60734010 *Packet Header*
6500C300 004760912 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 -------- -------- 0E3F89AC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F8AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F89AC 0E3F8D0C Test memory
0E3F89AC 0000000204 0E3F8AC 0E3F8AAAC 0E3F88AC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F89AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F8D0C Test memory
0E3F89AC 0000000204 0E3F8AC 0E3F8AAAC 0E3F88AC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F89AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F89AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F89AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F8D0C Test memory
0E3F8BAC 0000000204 0E3F89AC 0E3F8BAC 000 0E3F88AC 0E3F8BAC 0E3F8D0C 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:**

```plaintext
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
----------------

<table>
<thead>
<tr>
<th>Exec Count</th>
<th>Total CPU</th>
<th>Response Time (avg/max)</th>
<th>Queue Length (avg/max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>1</td>
<td>0/0</td>
<td>1/1</td>
</tr>
<tr>
<td>High</td>
<td>5</td>
<td>0/0</td>
<td>1/1</td>
</tr>
<tr>
<td>Normal</td>
<td>178</td>
<td>0/0</td>
<td>2/9</td>
</tr>
<tr>
<td>Low</td>
<td>15</td>
<td>0/0</td>
<td>2/3</td>
</tr>
</tbody>
</table>

Common Process Information
---------------------------

<table>
<thead>
<tr>
<th>PID</th>
<th>Name</th>
<th>Prio Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>AAA high-capacit M</td>
<td>New</td>
</tr>
<tr>
<td>133</td>
<td>RADIUS TEST CMD M</td>
<td>New</td>
</tr>
<tr>
<td>47</td>
<td>VNM DSPRM MAIN H</td>
<td>New</td>
</tr>
<tr>
<td>58</td>
<td>TurboACL M</td>
<td>New</td>
</tr>
<tr>
<td>97</td>
<td>IP Background M</td>
<td>New</td>
</tr>
<tr>
<td>99</td>
<td>CEF: IPv4 process L</td>
<td>New</td>
</tr>
<tr>
<td>112</td>
<td>X.25 Background M</td>
<td>New</td>
</tr>
<tr>
<td>117</td>
<td>LFDp Input Proc M</td>
<td>New</td>
</tr>
<tr>
<td>3</td>
<td>Init M</td>
<td>Old</td>
</tr>
</tbody>
</table>

CPU Intensive processes
------------------------

<table>
<thead>
<tr>
<th>PID</th>
<th>Total CPUs</th>
<th>Exec Count</th>
<th>Quant avg/max</th>
<th>Burst Count</th>
<th>Burst size avg(max)</th>
<th>Schedcall Count</th>
<th>Schedcall Count Per avg/max</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>820</td>
<td>6</td>
<td>136/236</td>
<td>1</td>
<td>24/24</td>
<td>18</td>
<td>887/15172</td>
</tr>
</tbody>
</table>

Priority Suspends
-----------------

<table>
<thead>
<tr>
<th>PID</th>
<th>Exec Count</th>
<th>Prio-Susps</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Latencies
---------

<table>
<thead>
<tr>
<th>PID</th>
<th>Exec Count</th>
<th>Latency avg/max</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>15192/15192</td>
</tr>
<tr>
<td>133</td>
<td>1</td>
<td>15192/15192</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>15192/15192</td>
</tr>
<tr>
<td>112</td>
<td>1</td>
<td>15192/15192</td>
</tr>
<tr>
<td>117</td>
<td>1</td>
<td>15192/15192</td>
</tr>
<tr>
<td>99</td>
<td>1</td>
<td>15172/15172</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>15172/15172</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>15172/15172</td>
</tr>
</tbody>
</table>

Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 00:00:00
Queue Statistics
-----------------

<table>
<thead>
<tr>
<th>Exec Count</th>
<th>Total CPU</th>
<th>Response Time (avg/max)</th>
<th>Queue Length (avg/max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>Normal</td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
</tbody>
</table>
Common Process Information

CPU Intensive processes

<table>
<thead>
<tr>
<th>PID</th>
<th>Total CPUms</th>
<th>Exec Count</th>
<th>avg/max Quant</th>
<th>Burst Count</th>
<th>avg/max(ms)</th>
<th>Schedcall Count</th>
<th>Per avg/max</th>
</tr>
</thead>
</table>

Priority Suspends

Latencies

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
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075563C 0x60755638
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075DD40 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
--- Burst 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
-------------------------

Embedded Resource Manager (ERM)

How to Configure Embedded Resource Manager

Embedded Resource Manager Configuration Guide, Cisco IOS Release 12.4T

---

**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: lospc

<table>
<thead>
<tr>
<th>Resource User</th>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init(ID: 0x1000001)</td>
<td>16777217</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Init</td>
</tr>
<tr>
<td>Scheduler(ID: 0x1000002)</td>
<td>16777218</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Scheduler</td>
</tr>
<tr>
<td>Dead(ID: 0x1000003)</td>
<td>16777219</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Dead</td>
</tr>
<tr>
<td>Interrupt(ID: 0x1000004)</td>
<td>16777220</td>
<td>15</td>
<td>13</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Interrupt</td>
</tr>
<tr>
<td>Memory RO RU(ID: 0x1000005)</td>
<td>16777221</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Memory RO RU</td>
</tr>
<tr>
<td>Chunk Manager(ID: 0x1000006)</td>
<td>16777222</td>
<td>13</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Chunk Manager</td>
</tr>
<tr>
<td>Load Meter(ID: 0x1000007)</td>
<td>16777223</td>
<td>2872</td>
<td>36029</td>
<td>79</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Check heaps(ID: 0x1000009)</td>
<td>16777225</td>
<td>352744</td>
<td>33446</td>
<td>10546</td>
<td>0.00%</td>
<td>0.20%</td>
</tr>
<tr>
<td>Pool Manager(ID: 0x100000A)</td>
<td>16777226</td>
<td>0</td>
<td>1</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Pool Manager</td>
</tr>
<tr>
<td>Buffer RO RU(ID: 0x100000B)</td>
<td>16777227</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Buffer RO RU</td>
</tr>
<tr>
<td>Timers(ID: 0x100000C)</td>
<td>16777228</td>
<td>2</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Timers</td>
</tr>
<tr>
<td>Serial Background(ID: 0x100000D)</td>
<td>16777229</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00% Serial Background</td>
</tr>
</tbody>
</table>
### Embedded Resource Manager (ERM)

- **How to Configure Embedded Resource Manager**

#### Embedded Resource Manager Configuration Guide, Cisco IOS Release 12.4T

<table>
<thead>
<tr>
<th>Resource User</th>
<th>Getbufs</th>
<th>Retbufs</th>
<th>Holding</th>
<th>RU Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>draco-oir-process:slot 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>draco-oir-process</td>
</tr>
<tr>
<td>SCP async: Draco-LC4</td>
<td>35849</td>
<td>243101</td>
<td>4294760044</td>
<td>SCP async: Draco-LC4</td>
</tr>
<tr>
<td>IFCOM Msg Hdlr</td>
<td>28</td>
<td>28</td>
<td>0</td>
<td>IFCOM Msg Hdlr</td>
</tr>
<tr>
<td>Exec</td>
<td>28</td>
<td>28</td>
<td>0</td>
<td>Exec</td>
</tr>
</tbody>
</table>

#### Resource User: AAA_SERVER_DEADTIME

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>2</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>Serial Background</td>
</tr>
</tbody>
</table>

#### Resource User: AAA_SERVER_DEADTIME(ID: 0x100000F)

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>AAA_SERVER_DEADTIME</td>
</tr>
</tbody>
</table>

#### Resource User: Policy Manager

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>Policy Manager</td>
</tr>
</tbody>
</table>

#### Resource User: Crash writer

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>Crash writer</td>
</tr>
</tbody>
</table>

#### Resource User: IPC Dynamic Cache

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11164</td>
<td>92859</td>
<td>120</td>
<td>0.00%</td>
<td>0.00%</td>
<td>EnvMon</td>
</tr>
</tbody>
</table>

#### Resource User: IPC Periodic Timer

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3004</td>
<td>79749</td>
<td>1</td>
<td>0.00%</td>
<td>0.00%</td>
<td>IPC Dynamic Cache</td>
</tr>
</tbody>
</table>

#### Resource User: IPC Deferred Port Closure

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>180082</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>IPC Periodic Timer</td>
</tr>
</tbody>
</table>

#### Resource User: IPC Seat Manager

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>97560</td>
<td>1408799</td>
<td>69</td>
<td>0.23%</td>
<td>0.02%</td>
<td>IPC Seat Manager</td>
</tr>
</tbody>
</table>

#### Resource User: IPC Session Service

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>3082</td>
<td>6</td>
<td>0.00%</td>
<td>0.00%</td>
<td>ARP Input</td>
</tr>
</tbody>
</table>

#### Resource User: EEM ED Syslog

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>EEM ED Syslog</td>
</tr>
</tbody>
</table>

#### Resource User: DDR Timers

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>180088</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>DDR Timers</td>
</tr>
</tbody>
</table>

#### Resource User: Dialer event

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>Dialer event</td>
</tr>
</tbody>
</table>

#### Resource User: Entity MIB API

<table>
<thead>
<tr>
<th>RUID Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>Res Usr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>28</td>
<td>16</td>
<td>1750</td>
<td>0.00%</td>
<td>0.00%</td>
<td>Entity MIB API</td>
</tr>
</tbody>
</table>

---
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        0.00%  0.00%  0.00%    Init
   Resource User: Scheduler(ID: 0x1000002)
   RUID Runtime(ms)   Invoked   uSecs   5Sec   1Min   5Min    Res Usr
16777218   0          0        0.00%  0.00%  0.00%    Scheduler
   Resource User: Dead(ID: 0x1000003)
   RUID Runtime(ms)   Invoked   uSecs   5Sec   1Min   5Min    Res Usr
16777219   0          0        0.00%  0.00%  0.00%    Dead
   Resource User: Interrupt(ID: 0x1000004)
   RUID Runtime(ms)   Invoked   uSecs   5Sec   1Min   5Min    Res Usr
16777220   0          0        0.00%  0.00%  0.00%    Interrupt
   Resource User: Memory RO RU(ID: 0x1000005)
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)
```
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
21197780  8950144  1229256  21742
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  14408  14408  14408  14408  14408  14408
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
  332       60      272  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
  16408  16408  16408  16408  16408  16408  16408  16408
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 enable</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td>- Enter your password if prompted.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> debug resource policy registration</td>
<td>Enables debugging on resource policy registration.</td>
</tr>
</tbody>
</table>

**Example:**

Router# debug resource policy registration

<table>
<thead>
<tr>
<th><strong>Step 3</strong> debug resource policy notification [owner resource-owner-name]</th>
<th>Enables notification debugging on ROs.</th>
</tr>
</thead>
</table>

**Example:**

Router# debug resource policy notification owner cpu

### 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
*Mar 3 09:35:58.304: resource_user_register: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0 name: usrr1
*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
```
Configuration Examples for Embedded Resource Manager

- Managing Resource Utilization by Defining Resource Policy Example, page 49
- Setting Expected Operating Ranges for Resource Owners Example, page 49
- Applying a Policy Example, page 54
- Setting a System Global Thresholding Policy for I/O Memory Example, page 55

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:

```config
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:

```config
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:

```config
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:

```config
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
```
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:

```sh
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:

```sh
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:

```sh
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
ranging 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:

```text
configure terminal
resource policy
policy user-local-pcl 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-pcl 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:

```text
configure terminal
resource policy
policy system-global-pcl 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-pcl 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:

```text
configure terminal
resource policy
policy per-user-global-pcl 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-pcl 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:

```text
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
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:

```bash
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:

```bash
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:

```bash
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:

```bash
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**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>Configuration fundamentals commands: complete command syntax, command modes,</td>
<td>Cisco IOS Configuration Fundamentals Command Reference</td>
</tr>
<tr>
<td>command history, defaults, usage guidelines, and examples</td>
<td></td>
</tr>
</tbody>
</table>
### Related Topic
| Document Title |  
|----------------|---
| Cisco IOS Network Management Command Reference |  
| Embedded Event Manager configuration tasks |  
| Memory Leak Detector |  

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-ERM-MIB.my</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature.</td>
<td>--</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
### 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Resource Manager</td>
<td>12.3(14)T 12.2(33)SRB 12.2(33)SB</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Feature Name

<table>
<thead>
<tr>
<th>Embedded Resource Manager MIB</th>
</tr>
</thead>
</table>

### Releases

<table>
<thead>
<tr>
<th>Embedded Resource Manager MIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.0(1)M 12.2(33)SRB</td>
</tr>
<tr>
<td>12.2(33)SB</td>
</tr>
</tbody>
</table>

### Feature Information

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`.

<table>
<thead>
<tr>
<th>Packet Memory Reclamation</th>
</tr>
</thead>
</table>

### Releases

<table>
<thead>
<tr>
<th>Packet Memory Reclamation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(6)T 12.2(33)SRE</td>
</tr>
</tbody>
</table>

### Feature Information

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 want 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.

---

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party
trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
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, page 61
- Prerequisites for ERM-MIB, page 61
- Information About ERM-MIB, page 61
- How to Configure ERM-MIB, page 71
- Configuration Examples for ERM-MIB, page 73
- Additional References, page 73
- Feature Information for ERM-MIB, page 74

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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, page 62
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>
<thead>
<tr>
<th>ERM Show MIB Objects</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>cermResOwnerTable</td>
<td>Obtains the details of all resource owners in the system.</td>
</tr>
<tr>
<td>cermResOwnerSubTypeTable</td>
<td>Obtains the details of the resource owner sub-types in the system.</td>
</tr>
<tr>
<td>cermResOwnerSubTypeThresholdTable</td>
<td>Obtains the details of the threshold value defined for each resource owner sub-type in the system.</td>
</tr>
<tr>
<td>cermResUserTypeTable</td>
<td>Obtains the details of the resource user types in the system.</td>
</tr>
<tr>
<td>cermResUserTable</td>
<td>Obtains the details of each resource user in the system.</td>
</tr>
<tr>
<td>cermResGroupTable</td>
<td>Obtains the details of each resource group in the system.</td>
</tr>
<tr>
<td>cermResGroupResUserTable</td>
<td>Obtains the details of resource users available in a specific resource group.</td>
</tr>
<tr>
<td>cermResOwnerResUserOrGroupTable</td>
<td>Obtains the details of all the resource owners, resource users, and group relationships defined in the system.</td>
</tr>
<tr>
<td>cermResOwnerResUserOrGroupThresholdTable</td>
<td>Obtains the details of the threshold value defined for each resource owner sub-type, resource user or resource group relationship in the system.</td>
</tr>
<tr>
<td>cermResUserTypeResOwnerTable</td>
<td>Obtains the details of resource owners present in a specific resource user type.</td>
</tr>
<tr>
<td>cermResMonitorTable</td>
<td>Obtains the details of resource monitors in the system.</td>
</tr>
<tr>
<td>cermResMonitorResOwnerResUserTable</td>
<td>Obtains the details of resource owners, resource users, and resource owner and resource user relationships that are monitored by a resource monitor.</td>
</tr>
</tbody>
</table>
### ERM Show MIB Objects

<table>
<thead>
<tr>
<th>ERM Show MIB Objects</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>cermResMonitorPolicyTable</td>
<td>Obtains the details of resource policies that are monitored by a resource monitor.</td>
</tr>
</tbody>
</table>

- Obtaining Information About Resource Owners, page 63
- Obtaining Sub-type Specific Information, page 63
- Obtaining Applied System Global Threshold Details, page 64
- Obtaining Information About a Resource User Type, page 64
- Obtaining Resource User-Specific Information, page 64
- Obtaining Information About Resource Groups, page 65
- Obtaining Information About Resource Users in a Particular Resource Group, page 65
- Obtaining Information About Resource Owner and User Relationships, page 65
- Obtaining Threshold Information About Each Resource Owner Sub-type and Resource User Relationship, page 66
- Obtaining Information About Resource Owners Present in a Resource User Type, page 66
- Obtaining Information About Resource Monitors, page 66
- Obtaining Resource Information About Resource Owner and User Relationships that are Monitored, page 66
- Obtaining Information About Resource Policies that are Monitored by a Resource Monitor, page 67

### 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 cermResUserId.

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, 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:

• cermResOwnerResUserOrGroupId
• cermResOwnerResUserOrGroupFlag
• 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, cermResOwnerResUserOrGroupId, cermResUserOrGroupThreshIsUserGlob, and cermResUserOrGroupThreshSeverity.

The cermResOwnerResUserOrGroupThresholdTable defines the following objects:

- cermResUserOrGroupThresholdIsUserGlob
- cermResUserOrGroupThresholdSeverity
- cermResUserOrGroupThresholdFlag
- 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, cermResUserId, 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>
<thead>
<tr>
<th>ERM Configuration MIB Objects</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>cermScalarsGlobalPolicyName (scalar object)</td>
<td>Identifies and indicates the global resource policy applied in the system.</td>
</tr>
<tr>
<td>cermConfigPolicyTable</td>
<td>Creates, modifies, or deletes a resource policy.</td>
</tr>
<tr>
<td>cermConfigPolicyResOwnerThreshTable</td>
<td>Configures threshold values and intervals for resource owner sub-types.</td>
</tr>
<tr>
<td>cermConfigResGroupTable</td>
<td>Creates or deletes a resource group.</td>
</tr>
<tr>
<td>cermConfigResGroupUserTable</td>
<td>Creates or deletes a user instance in a resource group.</td>
</tr>
<tr>
<td>cermConfigPolicyApplyTable</td>
<td>Applies an existing resource policy to a resource user or group.</td>
</tr>
</tbody>
</table>

- Verifying Whether a Global Resource Policy Is Applied in the System, page 68
- Creating Modifying or Deleting a Resource Policy, page 68
- Configuring Threshold Values and Intervals for Resource Owner Sub-types in a Resource Policy, page 68
- Creating or Deleting a Resource Group, page 69
- Creating or Deleting a User Instance in a Resource Group, page 69
- Applying an Existing Resource Policy to a Resource User or Group, page 69
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>
<thead>
<tr>
<th>ERM Notification MIB Objects</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>cermNotifsEnabled</td>
<td>Enables ERM notifications.</td>
</tr>
</tbody>
</table>
### ERM Notification MIB Objects and Purpose

<table>
<thead>
<tr>
<th>Object</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ciscoErmGlobalPolicyViolation</td>
<td>Specifies the type of notification received on global policy violation.</td>
</tr>
<tr>
<td>ciscoErmLocalPolicyViolation</td>
<td>Specifies the type of user-specific notification received on local policy violation.</td>
</tr>
</tbody>
</table>

- Controlling the Generation of Traps for ERM Policy Violation Notifications, page 70
- Receiving a Global Notification on Policy Violation, page 70
- Receiving a User-Specific Notification on Policy Violation, page 70

### 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
- cermResUserName
- cermResUserOrGroupThreshFlag
- cermNotifsThresholdIsUserGlob
- cermNotifsThresholdSeverity
- cermNotifsThresholdValue
- cermNotifsDirection
- cermNotifsPolicyName
How to Configure ERM-MIB

- Enabling ERM-MIB Notification Traps, page 71
- Configuring the Router to Send SNMP Notification Traps for ERM to a Host, page 72

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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> snmp-server enable traps resource-policy</td>
<td>Enables CISCO-ERM-MIB notifications.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# snmp-server enable traps resource-policy</td>
</tr>
<tr>
<td><strong>Step 4</strong> end</td>
<td>Returns the router to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# end</td>
</tr>
</tbody>
</table>
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.

- 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> show running-config [options]</td>
<td>Displays the running configuration to determine if an SNMP agent is already running.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show running-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> snmp-server host {hostname</td>
<td>ip-address} [vrf vrf-name] [traps</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# snmp-server host 209.165.201.30 traps version 2c priv mycommunitystring isis</td>
<td></td>
</tr>
</tbody>
</table>
**Command or Action** | **Purpose**
--- | ---
Step 5 end | Returns the router to privileged EXEC mode.

**Example:**
```
Router(config)# end
```

---

## Configuration Examples for ERM-MIB

- Configuring the Router to Send SNMP Notifications for ERM to a Host Example, page 73

## 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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>Embedded Resource Manager</td>
<td>Embedded Resource Manager</td>
</tr>
<tr>
<td>Network Management commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td>Cisco IOS Network Management Command Reference</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CISCO-ERM-MIB.my</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1902</td>
<td>Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Resource Manager (ERM)-MIB</td>
<td>12.2(33)SB 12.2(33)SRB 12.4(15)T</td>
<td>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: <code>snmp-server enable traps resource-policy</code></td>
</tr>
</tbody>
</table>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.