show odm-format

To display the schema of the spec file, use the **show odm-format** command in privileged EXEC mode.

show odm-format

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was modified. It was integrated into Cisco IOS Release 12.2(33)SRE.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

The **show odm-format** command displays the spec file structure in a fixed output that you can refer to in order to understand the spec file tag hierarchy.

Examples

The following example shows the fixed output from the **show odm-format** command:

```
Router# show odm-format

New Name Space ''
<NotARealTag> Either 0 or 1 allowed
  <ODMSpec> Exactly 1 required
    <Command> Exactly 1 required
      <Name> Exactly 1 required
      <AliasSet> Either 0 or 1 allowed
        <Alias> At least 1 required
    <OS> Either 0 or 1 allowed
    <DataModel> Exactly 1 required
      <Container> Exactly 1 required
        <Table> 0 or more is allowed
          <Header> At least 1 required
            <Option> 0 or more is allowed
          <EndOfTheTable> Either 0 or 1 allowed
        <Property> 0 or more is allowed
          <Option> 0 or more is allowed
        <Container> 0 or more is allowed
          <Table> 0 or more is allowed
            <Header> At least 1 required
              <Option> 0 or more is allowed
            <EndOfTheTable> Either 0 or 1 allowed
          <Property> 0 or more is allowed
            <Option> 0 or more is allowed
             <Container> 0 or more is allowed
          <Legends> 0 or more is allowed
            <Legend> At least 1 required
```

```
  <IgnorableLinesList> 0 or more is allowed
    <Line> At least 1 required
<Legends> 0 or more is allowed
  <Legend> At least 1 required
<IgnorableLinesList> 0 or more is allowed
  <Line> At least 1 required
```

The display from the **show odm-format** command is self-explanatory; see the "Usage Guidelines" section for more information.

Related Commands

Command	Description
show xsd-format	Generates XSD output for a command.

show platform hardware capacity

To display the capacities and utilizations for the hardware resources, use the **show platform hardware capacity** command in privileged EXEC mode.

show platform hardware capacity [*resource-type*]

Syntax Description

resource-type	(Optional) Hardware resource type; see the "Usage Guidelines" section for the valid values.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(18)SXF	Support for this command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXI	Support was added for the **ibc** and **rewrite-engine** keywords.

Usage Guidelines

The valid values for *resource-type* are as follows:

- **acl**—Displays the capacities and utilizations for ACL/QoS TCAM resources.
- **cpu**—Displays the capacities and utilizations for CPU resources.
- **eobc**—Displays the capacities and utilizations for Ethernet out-of-band channel resources.
- **fabric**—Displays the capacities and utilizations for Switch Fabric resources.
- **flash**—Displays the capacities and utilizations for Flash/NVRAM resources.
- **forwarding**—Displays the capacities and utilizations for Layer 2 and Layer 3 forwarding resources.
- **ibc**—Displays the capacities and utilizations for interboard communication resources.
- **interface**—Displays the capacities and utilizations for interface resources.
- **monitor**—Displays the capacities and utilizations for SPAN resources.
- **multicast**—Displays the capacities and utilizations for Layer 3 multicast resources.
- **netflow**—Displays the capacities and utilizations for NetFlow resources.
- **pfc**—Displays the capacities and utilizations for all the PFC resources including Layer 2 and Layer 3 forwarding, NetFlow, CPU rate limiters, and ACL/QoS TCAM resources.
- **power**—Displays the capacities and utilizations for power resources.
- **qos**—Displays the capacities and utilizations for QoS policer resources.
- **rate-limiter**—Displays the capacities and utilizations for CPU rate limiter resources.

- **rewrite-engine**—Displays the packet drop and performance counters of the central rewrite engine on supervisors and line cards. For detailed information, see the **show platform hardware capacity rewrite-engine** command documentation.
- **system**—Displays the capacities and utilizations for system resources.
- **vlan**—Displays the capacities and utilizations for VLAN resources.

The **show platform hardware capacity cpu** command displays the following information:

- CPU utilization for the last 5 seconds (busy time and interrupt time), the percentage of the last 1-minute average busy time, and the percentage of the last 5-minute average busy time.
- Processor memory total available bytes, used bytes, and percentage used.
- I/O memory total available bytes, used bytes, and percentage used.

The **show platform hardware capacity eobc** command displays the following information:

- Transmit and receive rate
- Packets received and packets sent
- Dropped received packets and dropped transmitted packets

The **show platform hardware capacity forwarding** command displays the following information:

- The total available entries, used entries, and used percentage for the MAC tables.
- The total available entries, used entries, and used percentage for the FIB TCAM tables. The display is done per protocol base.
- The total available entries, used entries, and used percentage for the adjacency tables. The display is done for each region in which the adjacency table is divided.
- The created entries, failures, and resource usage percentage for the NetFlow TCAM and ICAM tables.
- The total available entries and mask, used entries and mask, reserved entries and mask, and entries and mask used percentage for the ACL/QoS TCAM tables. The output displays the available, used, reserved, and used percentage of the labels. The output displays the resource of other hardware resources that are related to the ACL/QoS TCAMs (such as available, used, reserved, and used percentage of the LOU, ANDOR, and ORAND).
- The available, used, reserved, and used percentage for the CPU rate limiters.

The **show platform hardware capacity interface** command displays the following information:

- Tx/Rx drops—Displays the sum of transmit and receive drop counters on each online module (aggregate for all ports) and provides the port number that has the highest drop count on the module.
- Tx/Rx per port buffer size—Summarizes the port-buffer size on a per-module basis for modules where there is a consistent buffer size across the module.

The **show platform hardware capacity monitor** command displays the following SPAN information:

- The maximum local SPAN sessions, maximum RSPAN sessions, maximum ERSPAN sessions, and maximum service module sessions.
- The local SPAN sessions used/available, RSPAN sessions used/available, ERSPAN sessions used/available, and service module sessions used/available.

The **show platform hardware capacity multicast** command displays the following information:

- Multicast Replication Mode: ingress and egress IPv4 and IPv6 modes.
- The MET table usage that indicates the total used and the percentage used for each module in the system.

- The bidirectional PIM DF table usage that indicates the total used and the percentage used.

The **show platform hardware capacity system** command displays the following information:

- PFC operating mode (PFC Version: PFC3A, PFC3B, unknown, and so forth)
- Supervisor redundancy mode (RPR, RPR+, SSO, none, and so forth)
- Module-specific switching information, including the following information:
  - Part number (WS-SUP720-BASE, WS-X6548-RJ-45, and so forth)
  - Series (supervisor engine, fabric, CEF720, CEF256, dCEF256, or classic)
  - CEF Mode (central CEF, dCEF)

The **show platform hardware capacity vlan** command displays the following VLAN information:

- Total VLANs
- VTP VLANs that are used
- External VLANs that are used
- Internal VLANs that are used
- Free VLANs

Examples

This example shows how to display CPU capacity and utilization information for the route processor, the switch processor, and the LAN module in the Cisco 7600 series router:

```
Router# show platform hardware capacity cpu

CPU Resources
  CPU utilization: Module             5 seconds       1 minute       5 minutes
                   1  RP               0% /  0%            1%             1%
                   1  SP               5% /  0%            5%             4%
                   7                  69% /  0%           69%            69%
                   8                  78% /  0%           74%            74%
  Processor memory: Module   Bytes:       Total          Used           %Used
                    1  RP             176730048       51774704            29%
                    1  SP             192825092       51978936            27%
                    7                 195111584       35769704            18%
                    8                 195111584       35798632            18%
  I/O memory: Module         Bytes:       Total          Used           %Used
              1  RP                     35651584       12226672            34%
              1  SP                     35651584        9747952            27%
              7                         35651584        9616816            27%
              8                         35651584        9616816            27%
Router#
```

This example shows how to display EOBC-related statistics for the route processor, the switch processor, and the DFCs in the Cisco 7600 series router:

```
Router# show platform hardware capacity eobc

EOBC Resources
  Module                    Packets/sec     Total packets     Dropped packets
  1  RP      Rx:                    61            108982                   0
             Tx:                    37             77298                   0
  1  SP      Rx:                    34            101627                   0
             Tx:                    39            115417                   0
  7          Rx:                     5             10358                   0
             Tx:                     8             18543                   0
  8          Rx:                     5             12130                   0
             Tx:                    10             20317                   0
```

```
Router#
```

This example shows how to display the current and peak switching utilization:

```
Router# show platform hardware capacity fabric

Switch Fabric Resources
  Bus utilization: current is 100%, peak was 100% at 12:34 12mar45
  Fabric utilization:      ingress                    egress
      Module channel speed current peak                current peak
       1      0        20G   100%  100% 12:34 12mar45  100%    100% 12:34 12mar45

       1      1        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
       4      0        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
      13      0         8G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
Router#
```

This example shows how to display information about the total capacity, the bytes used, and the percentage that is used for the Flash/NVRAM resources present in the system:

```
Router# show platform hardware capacity flash

Flash/NVRAM Resources
  Usage: Module Device               Bytes:      Total          Used     %Used
         1  RP  bootflash:                     31981568      15688048       49%
         1  SP  disk0:                        128577536     105621504       82%
         1  SP  sup-bootflash:                 31981568      29700644       93%
         1  SP  const_nvram:                     129004           856        1%
         1  SP  nvram:                            391160         22065        6%
         7      dfc#7-bootflash:               15204352        616540        4%
         8      dfc#8-bootflash:               15204352             0        0%
Router#
```

This example shows how to display the capacity and utilization of the EARLs present in the system:

```
Router# show platform hardware capacity forwarding

L2 Forwarding Resources
           MAC Table usage:   Module  Collisions  Total       Used       %Used
                              6                0  65536         11          1%

             VPN CAM usage:                        Total       Used       %Used
                                                     512          0          0%
L3 Forwarding Resources
             FIB TCAM usage:                      Total       Used       %Used
                  72 bits (IPv4, MPLS, EoM)     196608          36          1%
                 144 bits (IP mcast, IPv6)       32768           7          1%

                     detail:      Protocol                   Used       %Used
                                  IPv4                         36          1%
                                  MPLS                          0          0%
                                  EoM                           0          0%

                                  IPv6                          4          1%
                                  IPv4 mcast                    3          1%
                                  IPv6 mcast                    0          0%

            Adjacency usage:                      Total       Used       %Used
                                                 1048576         175          1%

     Forwarding engine load:
                     Module       pps   peak-pps                    peak-time
                     6              8       1972  02:02:17 UTC Thu Apr 21 2005
```

```
Netflow Resources
          TCAM utilization:       Module       Created      Failed       %Used
                                  6                 1           0          0%
          ICAM utilization:       Module       Created      Failed       %Used
                                  6                 0           0          0%

                Flowmasks:   Mask#   Type        Features
                       IPv4:     0   reserved    none
                       IPv4:     1   Intf FulNAT_INGRESS NAT_EGRESS FM_GUARDIAN
                       IPv4:     2   unused      none
                       IPv4:     3   reserved    none

                       IPv6:     0   reserved    none
                       IPv6:     1   unused      none
                       IPv6:     2   unused      none
                       IPv6:     3   reserved    none

CPU Rate Limiters Resources
             Rate limiters:       Total         Used      Reserved       %Used
                   Layer 3            9            4             1         44%
                   Layer 2            4            2             2         50%
ACL/QoS TCAM Resources
  Key: ACLent - ACL TCAM entries, ACLmsk - ACL TCAM masks, AND - ANDOR,
       QoSent - QoS TCAM entries, QOSmsk - QoS TCAM masks, OR - ORAND,
       Lbl-in - ingress label, Lbl-eg - egress label, LOUsrc - LOU source,
       LOUdst - LOU destination, ADJ - ACL adjacency

  Module ACLent ACLmsk QoSent QoSmsk Lbl-in Lbl-eg LOUsrc LOUdst  AND  OR  ADJ
  6          1%     1%     1%     1%     1%     1%     0%     0%   0%  0%   1%

Router#
```

This example shows how to display the interboard communication resources:

```
Router# show platform hardware capacity ibc

IBC Resources
  Module                      Packets/sec     Total packets     Dropped packets
  1  RP      Rx:                        3           5001419                   0
             Tx:                        1           1943884                   0
Router#
```

This example shows how to display the interface resources:

```
Router# show platform hardware capacity interface

Interface Resources
  Interface drops:
    Module    Total drops:    Tx            Rx      Highest drop port:  Tx  Rx
    9                          0             2                           0  48

  Interface buffer sizes:
    Module                             Bytes:     Tx buffer           Rx buffer
         1                                            12345               12345
         5                                            12345               12345
Router#
```

This example shows how to display SPAN information:

```
Router# show platform hardware capacity monitor

SPAN Resources
  Source sessions: 2 maximum, 0 used
    Type                                    Used
    Local                                      0
```

```
    RSPAN source                                0
    ERSPAN source                               0
    Service module                              0
  Destination sessions: 64 maximum, 0 used
    Type                                    Used
    RSPAN destination                           0
    ERSPAN destination (max  24)                0
Router#
```

This example shows how to display the capacity and utilization of resources for Layer 3 multicast functionality:

```
Router# show platform hardware capacity multicast

L3 Multicast Resources
  IPv4 replication mode: ingress
  IPv6 replication mode: ingress
  Bi-directional PIM Designated Forwarder Table usage: 4 total, 0 (0%) used
  Replication capability: Module                                IPv4        IPv6
                          5                                   egress      egress
                          9                                  ingress     ingress
  MET table Entries: Module                                Total    Used    %Used
                     5                                     65526       6       0%
Router#
```

This example shows how to display information about the system power capacities and utilizations:

```
Router# show platform hardware capacity power

Power Resources
  Power supply redundancy mode: administratively combined
                                                      operationally combined
  System power: 1922W, 0W (0%) inline, 1289W (67%) total allocated
  Powered devices: 0 total
Router#
```

This example shows how to display the capacity and utilization of QoS policer resources per EARL in the Cisco 7600 series router:

```
Router# show platform hardware capacity qos

 QoS Policer Resources
  Aggregate policers: Module                      Total         Used     %Used
                      1                            1024          102       10%
                      5                            1024            1        1%
  Microflow policer configurations: Module        Total         Used     %Used
                                    1                64           32       50%
                                    5                64            1        1%
Router#
```

This example shows how to display information about the key system resources:

```
Router# show platform hardware capacity system

System Resources
  PFC operating mode: PFC3BXL
  Supervisor redundancy mode: administratively rpr-plus, operationally rpr-plus
  Switching Resources: Module   Part number              Series      CEF mode
                       5        WS-SUP720-BASE        supervisor          CEF
                       9        WS-X6548-RJ-45            CEF256          CEF
Router#
```

This example shows how to display VLAN information:

```
Router# show platform hardware capacity vlan
```

```
VLAN Resources
  VLANs: 4094 total, 10 VTP, 0 extended, 0 internal, 4084 free
Router#
```

Related Commands

Command	Description
show msfc	Displays MSFC information.
show platform	Displays platform information.
show platform hardware capacity rewrite-engine	Displays the packet drop and performance counters of the central rewrite engine on supervisors and line cards.

show platform hardware capacity rewrite-engine

To display the packet drop and performance counters of the central rewrite engine on supervisors and line cards, use the **show platform hardware capacity rewrite-engine** command in privileged EXEC mode.

show platform hardware capacity rewrite-engine {**drop** | **performance**} [**slot** *number*] [**rate** [*sample_interval*]] [**details**]

Syntax Description

drop	Displays the central rewrite engine drop counter values.
performance	Displays the central rewrite engine current performance counter values or the performance rate.
slot *number*	(Optional) Displays the counter values for the module in the specified slot. If no slot is specified, the counters are displayed for each slot.
rate [*sample_interval*]	(Optional) Displays the drop rate or rewrite rate for a sample interval in msec between 1 and 1000. The default interval is 50 msec.
details	(Optional) Displays each individual drop counter with its name and register ID number. This keyword is not available with the **performance** keyword.

Defaults

If the sample interval is not specified, the default interval is 50 msec.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SXI	Support for this command was introduced

Usage Guidelines

In the output of the **show platform hardware capacity rewrite-engine performance** command output, a value of "N/A" means the slot/channel has a rewrite engine, but does not support performance counters.

Examples

This example shows how to display the packet drop counters of the central rewrite engine in all installed supervisors and line cards:

```
Router# show platform hardware capacity rewrite-engine drop

slot channel   packet drops      total overruns
----+---------+-----------------+------------------+
1    0         0                 0
5    0         15440040          22
7    0         44                0
7    1         0                 0
```

This example shows how to display a detailed report of the packet drop counters of the module in slot 1:

```
Router# show platform hardware capacity rewrite-engine drop slot 1 details

slot channel drop_id description          packet drops     total overruns
----+-------+-------+--------------------+----------------+--------------+
1    0       0x5ED   DROP NON BPDU        0                0
1    0       0x5EB   DROP BPDU            0                0
1    1       0x5ED   DROP NON BPDU        0                0
1    1       0x5EB   DROP BPDU            0                0
```

This example shows how to display the packet drop counters of the module in slot 5 over the default sample interval of 50 msec:

```
Router# show platform hardware capacity rewrite-engine drop slot 5 rate

slot channel   drop rate [pps]        overrun [Y/N]
----+---------+---------------------+-------------+
5    0         120079                Y
```

This example shows how to display the packet drop counters of the module in slot 5 over a sample interval of 20 msec:

```
Router# show platform hardware capacity rewrite-engine drop slot 5 rate 20

slot channel   drop rate [pps]        overrun [Y/N]
----+---------+---------------------+-------------+
5    0         180000                N
```

This example shows how to display the performance counters of the central rewrite engine in all installed supervisors and line cards:

```
Router# show platform hardware capacity rewrite-engine performance

slot channel perf_id description          packets          total overruns
----+-------+-------+--------------------+----------------+--------------+
1    0       0x235   FAB RX 0             12870            0
1    0       0x237   FAB RX 1             0                0
1    0       0x27B   FAB TX 0             164              0
1    0       0x27F   FAB TX 1             0                0
1    0       0x350   REPLICATION ML3      0                0
1    0       0x351   REPLICATION ML2      0                0
1    0       0x352   RECIRC L2            0                0
1    0       0x353   RECIRC L3            0                0
1    0       0x34C   SPAN TX 0            0                0
1    0       0x34D   SPAN TX 1            0                0
1    0       0x34E   SPAN RX 0            0                0
1    0       0x34F   SPAN RX 1            0                0
1    0       0x354   SPAN TERMINATION     0                0
1    1       0x235   FAB RX 0             106065           0
1    1       0x237   FAB RX 1             0                0
1    1       0x27B   FAB TX 0             180806           0
1    1       0x27F   FAB TX 1             0                0
1    1       0x350   REPLICATION ML3      0                0
1    1       0x351   REPLICATION ML2      0                0
1    1       0x352   RECIRC L2            0                0
1    1       0x353   RECIRC L3            0                0
1    1       0x34C   SPAN TX 0            0                0
1    1       0x34D   SPAN TX 1            0                0
1    1       0x34E   SPAN RX 0            201              0
1    1       0x34F   SPAN RX 1            90201            0
1    1       0x354   SPAN TERMINATION     0                0
4    0       N/A
5    0       0xBE    FAB RX 0             181496           0
5    0       0xC0    FAB RX 1             0                0
5    0       0x112   FAB TX 0             992089           0
```

```
5    0       0x116   FAB TX 1             0               0
5    0       0x299   REPLICATION ML3      0               0
5    0       0x29A   REPLICATION ML2      0               0
5    0       0x29B   RECIRC L2            0               0
5    0       0x29C   RECIRC L3            0               0
5    0       0x295   SPAN TX 0            91166           0
5    0       0x296   SPAN TX 1            91313           0
5    0       0x297   SPAN RX 0            1               0
5    0       0x298   SPAN RX 1            1               0
5    0       0x29D   SPAN TERMINATION     0               0
```

This example shows how to display the performance counters of the module in slot 5:

```
Router# show platform hardware capacity rewrite-engine performance slot 5

slot channel perf_id description          packets          total overruns
----+-------+-------+--------------------+----------------+--------------+
5    0       0xBE    FAB RX 0             1330             0
5    0       0xC0    FAB RX 1             0                0
5    0       0x112   FAB TX 0             715253           0
5    0       0x116   FAB TX 1             0                0
5    0       0x299   REPLICATION ML3      0                0
5    0       0x29A   REPLICATION ML2      0                0
5    0       0x29B   RECIRC L2            0                0
5    0       0x29C   RECIRC L3            0                0
5    0       0x295   SPAN TX 0            1022             0
5    0       0x296   SPAN TX 1            1152             0
5    0       0x297   SPAN RX 0            1                0
5    0       0x298   SPAN RX 1            1                0
5    0       0x29D   SPAN TERMINATION     0                0
```

This example shows how to display the performance counters of the module in slot 5 over the default sample interval of 50 msec:

```
Router# show platform hardware capacity rewrite-engine performance slot 5 rate

slot channel perf_id description          packet rate[pps] overrun [Y/N]
----+-------+-------+--------------------+----------------+--------------+
5    0       0xBE    FAB RX 0             11680            N
5    0       0xC0    FAB RX 1             0                N
5    0       0x112   FAB TX 0             11680            N
5    0       0x116   FAB TX 1             0                N
5    0       0x299   REPLICATION ML3      0                N
5    0       0x29A   REPLICATION ML2      0                N
5    0       0x29B   RECIRC L2            0                N
5    0       0x29C   RECIRC L3            0                N
5    0       0x295   SPAN TX 0            5840             N
5    0       0x296   SPAN TX 1            5840             N
5    0       0x297   SPAN RX 0            0                N
5    0       0x298   SPAN RX 1            0                N
5    0       0x29D   SPAN TERMINATION     0                N
```

Related Commands

Command	Description
clear platform hardware capacity rewrite-engine counter	Clears the packet drop and performance counters of the central rewrite engine on supervisors and line cards.

show platform software evtmon

To display the internal data structures, for a Cisco ASR 1000 Series Embedded Services Processor (ESP), use the **show platform software evtmon** command.

show platform software evtmon {*FP*}

Syntax Description

bootflash	(Optional) Displays the internal data structures for the specified *bootflash*. Possible *bootflash* values are: • **f0**—Cisco ASR 1000 Series ESP slot 0 • **fp active**—Active Cisco ASR 1000 Series ESP

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced on the Cisco ASR 1000 Series Routers.

Examples

The following example displays the mounted file systems for active FP:

```
Router# show platform software evtmon fp active
Total IPv4 rules: 1
Total IPv6 rules: 0

   CG_ID Class_ID threshold interval  ACL_NAME
-----------------------------------------------------------
       1        1      100        1    myacl
```

show platform software trace level

To view the trace levels for a specific module, enter the **show platform software trace level** priviliged EXEC and diagnostic mode command.

show platform software trace level *process hardware-module slot*

Syntax Description

process	Specifies the process in which the tracing level is being set. Options currently include: • **chassis-manager**—The Chassis Manager process. • **cpp-control-process**—The CPP Control process • **cpp-driver**—The CPP driver process • **cpp-ha-server**—The CPP HA server process • **cpp-service-process**—The CPP service process • **forwarding-manager**—The Forwarding Manager process. • **host-manager**—The Host Manager process. • **interface-manager**—The Interface Manager process. • **ios**—The IOS process. • **logger**—The logging manager process • **pluggable-services**—The pluggable services process. • **shell-manager**—The Shell Manager process.
hardware-module	Specifies the hardware module where the process in which the trace level is being set is running. Options include: • **carrier-card**—The process is on a SPA Interface Processor (SIP). • **forwarding-processor**—The process is on an Embedded Services Processor (ESP). • **route-processor**—The process is on a Route Processor (RP).
slot	Specifies the slot of the *hardware-module*. Options include: • *number*—The number of the SIP slot of the hardware module where the trace level is being set. For instance, if you want to specify the SIP in SIP slot 2 of the router, enter 2 as the *number*. • *SIP-slot*/*SPA-bay*—The number of the SIP router slot and the number of the SPA bay of that SIP. For instance, if you want to specify the SPA in bay 2 of the SIP in router slot 3, enter 3/2. • **cpp active**—The Cisco Packet Processor (CPP) in the active ESP. • **cpp standby**—The CPP in the standby ESP.

- **f0**—The ESP in ESP slot 0.
- **f1**—The ESP in ESP slot 1
- **fp active**—The active ESP.
- **fp standby**—The standby ESP.
- **r0**—The RP in RP slot 0.
- **r1**—The RP in RP slot 1.
- **rp active**—The active RP.
- **rp standby**—The standby RP.

Command Modes

Privileged EXEC (#)
Diagnostic (diag)

Command Default

No default behavior or values.

The default tracing level on a Cisco ASR 1000 Series Router is critical. The tracing level can be changed using the **set platform software trace** command.

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced.

Usage Guidelines

This command is used to view trace levels. Trace levels, which determine which trace messages are generated, can be defined using the **set platform software trace** command.

Table 84 shows all of the trace levels that are available and provides descriptions of what types of messages are displayed with each set tracing level. This command is used to review these trace levels for various modules on the Aggregation Services routers.

Table 84 Tracing Levels and Descriptions

Trace Level	Level Number	Description
Emergency	0	The message is regarding an issue that makes the system unusable.
Alert	1	The message is regarding an action that must be taken immediately.
Critical	2	The message is regarding a critical condition. This is the default setting for every module on the Cisco ASR 1000 Series Routers.
Error	3	The message is regarding a system error.
Warning	4	The message is regarding a system warning
Notice	5	The message is regarding a significant issue, but the router is still working normally.

Table 84 Tracing Levels and Descriptions

Trace Level	Level Number	Description
Informational	6	The message is useful for informational purposes only.
Debug	7	The message provides debug-level output.
Verbose	8	All possible tracing messages are sent when the trace level is set to verbose.
Noise	-	The noise tracing level will always send all possible trace messages for the module. The noise level is always equal to the highest possible tracing level. Even if a future enhancement to this command introduces a higher tracing level, the noise level will become equal to the level of that new enhancement.

Examples

In the following example, the **show platform software trace level** command is used to view the tracing levels of the Forwarding Manager processes on the active RP:

```
Router# show platform software trace level forwarding-manager rp active
Module Name                      Trace Level
-----------------------------------------------
acl                              Error
binos                            Error
binos/brand                      Error
bipc                             Error
btrace                           Error
cce                              Error
cdllib                           Error
cef                              Error
chasfs                           Error
chasutil                         Error
erspan                           Error
ess                              Error
ether-channel                    Error
evlib                            Error
evutil                           Error
file_alloc                       Error
fman_rp                          Error
fpm                              Error
fw                               Error
icmp                             Error
interfaces                       Error
iosd                             Error
ipc                              Error
ipclog                           Error
iphc                             Error
ipsec                            Error
mgmte-acl                        Error
mlp                              Error
mqipc                            Error
nat                              Error
nbar                             Error
netflow                          Error
om                               Error
peer                             Error
```

```
qos                              Error
route-map                        Error
sbc                              Error
services                         Error
sw_wdog                          Error
tdl_acl_config_type              Error
tdl_acl_db_type                  Error
tdl_cdl_message                  Error
tdl_cef_config_common_type       Error
tdl_cef_config_type              Error
tdl_dpidb_config_type            Error
tdl_fman_rp_comm_type            Error
tdl_fman_rp_message              Error
tdl_fw_config_type               Error
tdl_hapi_tdl_type                Error
tdl_icmp_type                    Error
tdl_ip_options_type              Error
tdl_ipc_ack_type                 Error
tdl_ipsec_db_type                Error
tdl_mcp_comm_type                Error
tdl_om_type                      Error
tdl_ui_type                      Error
tdl_urpf_config_type             Error
tdllib                           Error
trans_avl                        Error
uihandler                        Error
uipeer                           Error
uistatus                         Error
urpf                             Error
vista                            Error
```

Related Commands

Command	Description
set platform software trace	Sets the trace level for a specific module.
show platform software trace message	Displays the trace message for a specified module.

show platform software trace message

To display trace messages for a module, enter the **show platform software trace message** command in privileged EXEC mode or diagnostic mode.

show platform software trace message *process hardware-module slot*

Syntax Description

process	The process in which the tracing level is being set. The following keywords are available: • **chassis-manager**—The Chassis Manager process. • **cpp-control-process**—The Cisco packet processor (CPP) Control process. • **cpp-driver**—The CPP driver process. • **cpp-ha-server**—The CPP high availability (HA) server process. • **cpp-service-process**—The CPP service process. • **forwarding-manager**—The Forwarding Manager process. • **host-manager**—The Host Manager process. • **interface-manager**—The Interface Manager process. • **ios**—The Cisco IOS process. • **logger**—The logging manager process. • **pluggable-services**—The pluggable services process. • **shell-manager**—The Shell Manager process.
hardware-module	Tthe hardware module where the process whose trace level is being set is running. The following keywords are available: • **carrier-card**—The process is on an SPA Interface Processor (SIP). • **forwarding-processor**—The process is on an embedded services processor (ESP). • **route-processor**—The process is on an route processor (RP).
slot	The slot of the hardware module. Options are as follows: • *number*—The number of the SIP slot of the hardware module where the trace level is being set. For instance, if you want to specify the SIP in SIP slot 2 of the router, enter **2**. • *SIP-slot*/*SPA-bay*—The number of the SIP router slot and the number of the shared port adapter (SPA) bay of that SIP. For instance, if you want to specify the SPA in bay 2 of the SIP in router slot 3, enter **3/2**. • **cpp active**—The CPP in the active ESP. • **cpp standby**—The CPP in the standby ESP. • **f0**—The ESP in ESP slot 0. • **f1**—The ESP in ESP slot 1 • **fp active**—The active ESP. • **fp standby**—The standby ESP.

- **r0**—The RP in RP slot 0.
- **r1**—The RP in RP slot 1.
- **rp active**—The active RP.
- **rp standby**—The standby RP.

Command Modes

Privileged EXEC (#)
Diagnostic (diag)

Command History

Release	Modification
Cisco IOS XE Release 2.1	This command was introduced.
12.2(33)XND	This command was modified. The command output displays the truncated traceback message also.

Usage Guidelines

The **show platform software trace message** command is used to display trace messages from an in-memory message ring of a module's process that keeps a condensed historical record of all messages. Although all messages are saved in a trace log file unmodified, only the first 128 bytes of a message are saved in the message ring. The size limitation does not apply to the traceback portion of a message.

Examples

The following example shows how to display the trace messages for the Host Manager process in RP slot 0 using the **show platform software trace message** command:

```
Router# show platform software trace message host-manager R0

08/23 12:09:14.408 [uipeer]: (info): Looking for a ui_req msg
08/23 12:09:14.408 [uipeer]: (info): Start of request handling for con 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Accepted connection for 14 as 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Received new connection 0x100a61c8 on descriptor 14
08/23 12:09:14.398 [uipeer]: (info): Accepting command connection on listen fd 7
08/23 11:53:57.440 [uipeer]: (info): Going to send a status update to the shell manager in
slot 0
08/23 11:53:47.417 [uipeer]: (info): Going to send a status update to the shell manager in
slot 0
```

The following example shows a truncated message that has a traceback. The truncated portion of the message is indicated by an ellipsis (…):

```
03/02 15:47:44.002 [errmsg]: (ERR): %EVENTLIB-3-TIMEHOG: read asyncon 0x100a9260: 60618ms,
Traceback=1#862f8780825f93a618ecd9 ...Traceback=1#862f8780825f93a618ecd9dd48b3be96
evlib:FCAF000+CC00 evlib:FCAF000+A6A8 evutil:FFCA000+ADD0 evutil:FFCA000+5A80
evutil:FFCA000+A68C uipeer:FF49000+10AFC evlib:FCAF000+D28C evlib:FCAF000+F4C4
:10000000+1B24C c:EF44000+1D078 c:EF44000+1D220
```

Related Commands

Command	Description
set platform software trace	Sets the trace level for a specific module.
show platform software trace levels	Displays trace levels for a module.

show processes cpu

To display detailed CPU utilization statistics (CPU use per process) when Cisco IOS or Cisco IOS Software Modularity images are running, use the **show processes cpu** command in user EXEC or privileged EXEC mode.

Cisco IOS Software

show processes cpu [**history** [**table**] | **sorted** [**1min** | **5min** | **5sec**]]

Cisco IOS Software Modularity

show processes cpu [**detailed** [*process-id* | *process-name*] | **history**]

Syntax Description

history	(Optional) Displays CPU history in a graph format.
table	(Optional) Displays CPU history in a table format.
sorted	(Optional) For Cisco IOS images only. Displays CPU utilization sorted by percentage.
1min	(Optional) Sorts CPU utilization based on 1 minute utilization.
5min	(Optional) Sorts CPU utilization based on 5 minutes utilization.
5sec	(Optional) Sorts CPU utilization based on 5 seconds utilization.
detailed	(Optional) For Cisco IOS Software Modularity images only. Displays more detailed information about Cisco IOS processes (not for POSIX processes).
process-id	(Optional) For Cisco IOS Software Modularity images only. Process identifier.
process-name	(Optional) For Cisco IOS Software Modularity images only. Process name.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.0	This command was introduced.
12.2(2)T	This command was modified. The **history** keyword was added.
12.3(8)	This command was enhanced to display Address Resolution Protocol (ARP) output.
12.3(14)T	This command was enhanced to display ARP output.
12.2(18)SXF4	This command was enhanced to support Cisco IOS Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.2(33)SCB3	This command was integrated into Cisco IOS Release 12.2(33)SCB3. Support was added for Cisco uBR10012 and uBR7200 routers.

Release	Modification
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
15.0(1)M	This command was modified. The output was modified to display the CPU time in microseconds that the process has used.

Usage Guidelines

Cisco IOS Software

If you use the optional **history** keyword, three graphs are displayed for Cisco IOS images:

- CPU utilization for the last 60 seconds
- CPU utilization for the last 60 minutes
- CPU utilization for the last 72 hours

Maximum usage is measured and recorded every second; average usage is calculated on periods of more than one second. Consistently high CPU utilization over an extended period indicates a problem. Use the **show processes cpu** command to troubleshoot. Also, you can use the output of this command in the Cisco Output Interpreter tool to display potential issues and fixes. Output Interpreter is available to registered users of Cisco.com who are logged in and have Java Script enabled.

For a list of system processes, go to http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_tech_note09186a00800a65d0.shtml.

Cisco IOS Software Modularity

Cisco IOS Software Modularity images display only one graph that shows the CPU utilization for the last 60 minutes. The horizontal axis shows times (for example, 0, 5, 10, 15 minutes), and the vertical axis shows total percentage of CPU utilization (0 to 100 percent).

Examples

Example output varies between Cisco IOS software images and Cisco IOS Software Modularity software images. The following sections show output examples for each image:

- Cisco IOS Software
- Cisco IOS Software Modularity

Cisco IOS Software

The following is sample output from the **show processes cpu** command without keywords:

```
Router# show processes cpu

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
 PID Runtime(uS)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1        4000          67         59  0.00%  0.00%  0.00%   0 Chunk Manager
   2        4000      962255          0  0.00%  0.00%  0.00%   0 Load Meter
   3           0           1          0  0.00%  0.00%  0.00%   0 cpf_process_tp
   4           0           1          0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   5   586520704      732013       6668  0.00%  0.11%  0.08%   0 Check heaps
   6        4000         991          4  0.00%  0.00%  0.00%   0 Pool Manager
   7           0           1          0  0.00%  0.00%  0.00%   0 DiscardQ Backg
   8           0           2          0  0.00%  0.00%  0.00%   0 Timers
   9           0           2          0  0.00%  0.00%  0.00%   0 ATM AutoVC Per
  10           0           2          0  0.00%  0.00%  0.00%   0 ATM VC Auto Cr
  11  2154956000     4809201        448  0.00%  0.03%  0.03%   0 EnvMon
```

```
 PID Runtime(uS)      Invoked      uSecs   5Sec   1Min   5Min TTY Process
  12           0            1          0  0.00%  0.00%  0.00%   0 OIR Handler
  13           0            1          0  0.00%  0.00%  0.00%   0 Crash writer
  14           0            1          0  0.00%  0.00%  0.00%   0 IPC Process le
  15           0        80189          0  0.00%  0.00%  0.00%   0 IPC Dynamic Ca
  16           0            1          0  0.00%  0.00%  0.00%   0 IPC Zone Manag
  17           0       962246          0  0.00%  0.00%  0.00%   0 IPC Service No
  18           0      4698177          0  0.00%  0.00%  0.00%   0 IPC Periodic T
  19           0      4698177          0  0.00%  0.00%  0.00%   0 IPC Deferred P
  20           0            1          0  0.00%  0.00%  0.00%   0 IPC Seat Manag
  21           0            1          0  0.00%  0.00%  0.00%   0 IPC Seat Contr
  22           0       962246          0  0.00%  0.00%  0.00%   0 IPC Loadometer
<snip>
```

The following is sample output of the one-hour portion of the output. The Y-axis of the graph is the CPU utilization. The X-axis of the graph is the increment within the time period displayed in the graph. This example shows the individual minutes during the previous hour. The most recent measurement is on the left of the X-axis.

```
Router# show processes cpu history

!--- One minute output omitted

6665776865756676676666667667677676766666766767767666566667
6378016198993513709771991443732358689932740858269643922613
100
90
80         *  *                     * *     *  * *  *
70  * * ***** *  ** ***** ***  **** ******  *  *******     * *
60  #***##*##*#***######*#*########*#*###*#*#*##*#*##*#*##*****#
50  ############################################################
40  ############################################################
30  ############################################################
20  ############################################################
10  ############################################################
   0....5....1....1....2....2....3....3....4....4....5....5....
             0    5    0    5    0    5    0    5    0    5
              CPU% per minute (last 60 minutes)
              * = maximum CPU% # = average CPU%

!--- 72-hour output omitted
```

The top two rows, read vertically, display the highest percentage of CPU utilization recorded during the time increment. In this example, the CPU utilization for the last minute recorded is 66 percent. The device may have reached 66 percent only once during that minute, or it may have reached 66 percent multiple times. The device records only the peak reached during the time increment and the average over the course of that increment.

The following is sample output from the **show processes cpu** command on a Cisco uBR10012 router:

```
Router# show processes cpu

CPU utilization for five seconds: 2%/0%; one minute: 2%; five minutes: 2%
 PID Runtime(us)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1           8       471         16  0.00%  0.00%  0.00%   0 Chunk Manager
   2           4       472          8  0.00%  0.00%  0.00%   0 Load Meter
   3           0         1          0  0.00%  0.00%  0.00%   0 IPC 0x50000 Vers
   4           0        10          0  0.00%  0.00%  0.00%   0 C10K Card Event
   5           0        65          0  0.00%  0.00%  0.00%   0 Retransmission o
   6           0         5          0  0.00%  0.00%  0.00%   0 IPC ISSU Dispatc
   7        5112       472      10830  0.63%  0.18%  0.18%   0 Check heaps
   8           0         1          0  0.00%  0.00%  0.00%   0 Pool Manager
   9           0         2          0  0.00%  0.00%  0.00%   0 Timers
```

```
 10           0          2           0  0.00%  0.00%  0.00%   0 Serial Backgroun
 11           0        786           0  0.00%  0.00%  0.00%   0 WBCMTS process
 12           0          1           0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
 13           0          1           0  0.00%  0.00%  0.00%   0 Policy Manager
 14           0          1           0  0.00%  0.00%  0.00%   0 Crash writer
 15           0          1           0  0.00%  0.00%  0.00%   0 RO Notify Timers
 16           0          1           0  0.00%  0.00%  0.00%   0 RMI RM Notify Wa
 17           0       2364           0  0.00%  0.00%  0.00%   0 Facility Alarm
 18           0         41           0  0.00%  0.00%  0.00%   0 IPC Dynamic Cach
```

The following is sample output from the **show processes cpu** command that shows an ARP probe process:

```
Router# show processes cpu | include ARP

17       38140    389690          97  0.00%  0.00%  0.00%   0 ARP Input
36           0         1           0  0.00%  0.00%  0.00%   0 IP ARP Probe
40           0         1           0  0.00%  0.00%  0.00%   0 ATM ARP INPUT
80           0         1           0  0.00%  0.00%  0.00%   0 RARP Input
114          0         1           0  0.00%  0.00%  0.00%   0 FR ARP
```

Table 85 describes the fields shown in the output.

Table 85 show processes cpu Field Descriptions

Field	Description
CPU utilization for five seconds	CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
one minutes	CPU utilization for the last minute.
five minutess	CPU utilization for the last 5 minutes.
PID	Process ID.
Runtime (us)	CPU time that the process has used (in microseconds).
Invoked	Number of times that the process has been invoked.
uSecs	Microseconds of CPU time for each process invocation.
5Sec	CPU utilization by task in the last 5 seconds.
1Min	CPU utilization by task in the last minute.
5Min	CPU utilization by task in the last 5 minutes.
TTY	Terminal that controls the process.
Process	Name of the process.

Note Because platforms have a 4- to 8-microsecond clock resolution, run times are considered reliable only after several invocations or a reasonable, measured run time.

Cisco IOS Software Modularity

The following is sample output from the **show processes cpu** command when a Software Modularity image is running:

```
Router# show processes cpu

Total CPU utilization for 5 seconds: 99.6%; 1 minute: 98.5%; 5 minutes: 85.3%
PID       5Sec    1Min     5Min Process
```

```
1          0.0%    0.1%     0.8% kernel
3          0.0%    0.0%     0.0% qdelogger
4          0.0%    0.0%     0.0% devc-pty
6          0.7%    0.2%     0.1% devc-ser2681
7          0.0%    0.0%     0.0% dumper.proc
4104       0.0%    0.0%     0.0% pipe
8201       0.0%    0.0%     0.0% mqueue
8202       0.0%    0.0%     0.0% fsdev.proc
8203       0.0%    0.0%     0.0% flashfs_hes_slot1.proc
8204       0.0%    0.0%     0.0% flashfs_hes_slot0.proc
8205       0.0%    0.0%     0.0% flashfs_hes_bootflash.proc
8206       0.0%    0.0%     0.0% dfs_disk2.proc
8207       0.0%    0.0%     0.0% dfs_disk1.proc
8208       0.0%    0.0%     0.0% dfs_disk0.proc
8209       0.0%    0.0%     0.0% ldcache.proc
8210       0.0%    0.0%     0.0% watchdog.proc
8211       0.0%    0.0%     0.0% syslogd.proc
8212       0.0%    0.0%     0.0% name_svr.proc
8213       0.0%    0.1%     0.0% wdsysmon.proc
8214       0.0%    0.0%     0.0% sysmgr.proc
8215       0.0%    0.0%     0.0% kosh.proc
12290      0.0%    0.0%     0.0% chkptd.proc
12312      0.0%    0.0%     0.0% sysmgr.proc
12313      0.0%    0.0%     0.0% syslog_dev.proc
12314      0.0%    0.0%     0.0% itrace_exec.proc
12315      0.0%    0.0%     0.0% packet.proc
12316      0.0%    0.0%     0.0% installer.proc
12317     29.1%   28.5%    19.6% ios-base
12318      0.0%    0.0%     0.0% fh_fd_oir.proc
12319      0.0%    0.0%     0.1% fh_fd_cli.proc
12320      0.0%    0.0%     0.0% fh_metric_dir.proc
12321      0.0%    0.0%     0.0% fh_fd_snmp.proc
12322      0.0%    0.0%     0.0% fh_fd_none.proc
12323      0.0%    0.0%     0.0% fh_fd_intf.proc
12324     48.5%   48.5%    35.8% iprouting.iosproc
12325      0.0%    0.0%     0.0% fh_fd_timer.proc
12326      0.0%    0.0%     0.0% fh_fd_ioswd.proc
12327      0.0%    0.0%     0.0% fh_fd_counter.proc
12328      0.0%    0.0%     0.0% fh_fd_rf.proc
12329      0.0%    0.0%     0.0% fh_server.proc
12330      0.0%    0.0%     0.0% cdp2.iosproc
12331      0.0%    0.0%     0.0% fh_policy_dir.proc
12332      0.0%    0.0%     0.0% ipfs_daemon.proc
12333      0.0%    0.0%     0.0% raw_ip.proc
12334      0.0%    0.0%     0.0% inetd.proc
12335     19.1%   20.4%    12.6% tcp.proc
12336      0.0%    0.0%     0.0% udp.proc
```

Table 86 describes the significant fields shown in the display.

Table 86 show processes cpu (Software Modularity) Field Descriptions

Field	Description
Total CPU utilization for five seconds	Total CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
1 minute	CPU utilization for the last minute.
5 minutes	CPU utilization for the last 5 minutes.
PID	Process ID.

Table 86 show processes cpu (Software Modularity) Field Descriptions (continued)

Field	Description
5Sec	Percentage of CPU time spent at the interrupt level for this process during the last five seconds.
1Min	Percentage of CPU time spent at the interrupt level for this process during the last minute.
5Min	Percentage of CPU time spent at the interrupt level for this process during the last five minutes.
Process	Process name.

The following is partial sample output from the **show processes cpu** command with the **detailed** keyword when a Software Modularity image is running:

```
Router# show processes cpu detailed

Total CPU utilization for 5 seconds: 99.6%; 1 minute: 99.3%; 5 minutes: 88.6%
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
1         0.0%    0.7%     0.7% kernel                                 8.900
      1   0.4%    0.7%    11.4% [idle thread]          0  Ready        2m28s
      2   0.0%    0.0%     0.0%                       63  Receive      0.000
      3   0.0%    0.0%     0.0%                       10  Receive      0.000
      4   0.0%    0.0%     0.1%                       11  Receive      1.848
      5   0.0%    0.0%     0.0%                       63  Receive      0.000
.
.
.
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
8214      0.0%    0.0%     0.0% sysmgr.proc                            0.216
      1   0.0%    0.0%     0.0%                       10  Receive      0.132
      2   0.0%    0.0%     0.0%                       10  Sigwaitin    0.000
      3   0.0%    0.0%     0.0%                       10  Receive      0.004
      4   0.0%    0.0%     0.0%                       10  Receive      0.000
      5   0.0%    0.0%     0.0%                       10  Receive      0.000
      6   0.0%    0.0%     0.0%                       10  Receive      0.004
      7   0.0%    0.0%     0.0%                       10  Receive      0.000
      8   0.0%    0.0%     0.0%                       10  Receive      0.000
      9   0.0%    0.0%     0.0%                       10  Receive      0.000
     10   0.0%    0.0%     0.0%                       10  Receive      0.000
     11   0.0%    0.0%     0.0%                       10  Receive      0.000
     12   0.0%    0.0%     0.0%                       10  Receive      0.000
     13   0.0%    0.0%     0.0%                       10  Receive      0.028
     14   0.0%    0.0%     0.0%                       10  Receive      0.040
     15   0.0%    0.0%     0.0%                       10  Receive      0.000
     16   0.0%    0.0%     0.0%                       10  Receive      0.000
     17   0.0%    0.0%     0.0%                       10  Receive      0.004
     18   0.0%    0.0%     0.0%                       10  Receive      0.000
     19   0.0%    0.0%     0.0%                       10  Receive      0.000
     20   0.0%    0.0%     0.0%                       10  Receive      0.000
     21   0.0%    0.0%     0.0%                       10  Receive      0.004
     22   0.0%    0.0%     0.0%                       10  Receive      0.000
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
8215      0.0%    0.0%     0.0% kosh.proc                              0.044
      1   0.0%    0.0%     0.0%                       10  Reply        0.044
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
12290     0.0%    0.0%     0.0% chkptd.proc                            0.080
      1   0.0%    0.0%     0.0%                       10  Receive      0.080
      2   0.0%    0.0%     0.0%                       10  Receive      0.000
PID/TID   5Sec    1Min     5Min Process             Prio  STATE        CPU
```

```
12312     0.0%    0.0%    0.0% sysmgr.proc                              0.112
      1   0.0%    0.0%    0.0%                        10  Receive       0.112
      2   0.0%    0.0%    0.0%                        10  Sigwaitin     0.000
PID/TID   5Sec    1Min    5Min Process             Prio  STATE         CPU
12316     0.0%    0.0%    0.0% installer.proc                           0.072
      1   0.0%    0.0%    0.0%                        10  Receive       0.000
      3   0.0%    0.0%    0.0%                        10  Nanosleep     0.000
      4   0.0%    0.0%    0.0%                        10  Sigwaitin     0.000
      6   0.0%    0.0%    0.0%                        10  Receive       0.000
Process sbin/ios-base, type IOS, PID = 12317
CPU utilization for five seconds: 12%/9%; one minute: 13%; five minutes: 10%
Task  Runtime(us)  Invoked  uSecs    5Sec   1Min   5Min TTY Task Name
   1         219     1503    145   0.00%  0.00%  0.00%   0 Hot Service Task
   2       23680    42384    558   2.39%  6.72%  4.81%   0 Service Task
   3        6104    11902    512   3.51%  1.99%  1.23%   0 Service Task
   4        1720     5761    298   1.91%  0.90%  0.39%   0 Service Task
   5           0        5      0   0.00%  0.00%  0.00%   0 Chunk Manager
   6           0        1      0   0.00%  0.00%  0.00%   0 Connection Mgr
   7           4      106     37   0.00%  0.00%  0.00%   0 Load Meter
   8        6240     7376    845   0.23%  0.15%  0.55%   0 Exec
   9         379       62   6112   0.00%  0.07%  0.04%   0 Check heaps
  10           0        1      0   0.00%  0.00%  0.00%   0 Pool Manager
  11           3        2   1500   0.00%  0.00%  0.00%   0 Timers
  12           0        1      0   0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  13           0        2      0   0.00%  0.00%  0.00%   0 AAA high-capacit
  14         307      517    593   0.00%  0.05%  0.03%   0 EnvMon
  15           0        1      0   0.00%  0.00%  0.00%   0 OIR Handler
  16         283       58   4879   0.00%  0.04%  0.02%   0 ARP Input
  17           0        2      0   0.00%  0.00%  0.00%   0 Serial Backgroun
  18           0       81      0   0.00%  0.00%  0.00%   0 ALARM_TRIGGER_SC
  19           0        2      0   0.00%  0.00%  0.00%   0 DDR Timers
  20           0        2      0   0.00%  0.00%  0.00%   0 Dialer event
  21           4        2   2000   0.00%  0.00%  0.00%   0 Entity MIB API
  22           0       54      0   0.00%  0.00%  0.00%   0 Compute SRP rate
  23           0        9      0   0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  24           0        1      0   0.00%  0.00%  0.00%   0 IPC Zone Manager
  25           0        1      0   0.00%  0.00%  0.00%   0 IPC Punt Process
  26           4      513      7   0.00%  0.00%  0.00%   0 IPC Periodic Tim
  27          11      513     21   0.00%  0.00%  0.00%   0 IPC Deferred Por
  28           0        1      0   0.00%  0.00%  0.00%   0 IPC Seat Manager
  29          83     1464     56   0.00%  0.00%  0.00%   0 EEM ED Syslog
.
.
.
```

Table 87 describes the significant fields shown in the display.

Table 87 show processes cpu detailed (Software Modularity) Field Descriptions

Field	Description
Total CPU utilization for five seconds	Total CPU utilization for the last 5 seconds. The second number indicates the percent of CPU time spent at the interrupt level.
1 minute	CPU utilization for the last minute.
5 minutes	CPU utilization for the last 5 minutes.
PID/TID	Process ID or task ID.
5Sec	Percentage of CPU time spent at the interrupt level for this process during the last five seconds.

Table 87 show processes cpu detailed (Software Modularity) Field Descriptions (continued)

Field	Description
1Min	Percentage of CPU time spent at the interrupt level for this process during the last minute.
5Min	Percentage of CPU time spent at the interrupt level for this process during the last five minutes.
Process	Process name.
Prio	Priority level of the process.
STATE	Current state of the process.
CPU	CPU utilization of the process in minutes and seconds.
type	Type of process; can be either IOS or POSIX.
Task	Task sequence number.
Runtime(us)	CPU time that the process has used (in microseconds).
Invoked	Number of times that the process has been invoked.
uSecs	Microseconds of CPU time for each process invocation.
5Sec	CPU utilization by task in the last 5 seconds.
1Min	CPU utilization by task in the last minute.
5Min	CPU utilization by task in the last 5 minutes.
TTY	Terminal that controls the process.
Task Name	Task name.

Related Commands

Command	Description
show processes	Displays information about active processes.
show processes memory	Displays the amount of system memory used per system process.

show processes cpu autoprofile hog

To see the CPUHOG profile data, use the **show processes cpu autoprofile hog** command in user EXEC or privileged EXEC mode.

show processes cpu autoprofile hog

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show processes cpu autoprofile hog** command:

```
Router# show processes cpu autoprofile hog

0x6075DD40 0x60755638
0x6075DD24 0x60755638
0x6075563C 0x60755638
0x60755638 0x60755638
0x60755638 0x60755638
0x6075DD10 0x60755638
0x6075DD40 0x60755638
0x6075DD40 0x60755638
0x6075563C 0x60755638
0x6075DCE0 0x60755638
0x6075DD44 0x60755638
.
.
.
0x6075DCCC 0x60755638
0x6075DCDC 0x60755638
0x6075563C 0x60755638
0x6075DD3C 0x60755638
0x6075DD20 0x60755638
0x6075DD58 0x60755638
0x6075DD1C 0x60755638
0x6075DD10 0x60755638
0x6075DCDC 0x60755638
0x6075DCF8 0x60755638
```

Related Commands

Command	Description
processes cpu autoprofile hog	Enables automatic CPU profiling.

show processes cpu autoprofile process

To display the CPU process buffer profile data, use the **show processes cpu autoprofile process** command in user EXEC or privileged EXEC mode.

show processes cpu autoprofile process

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.1	This command was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.

Examples

The following is sample output from the **show processes cpu autoprofile process** command. The output is self-explanatory.

```
Router# show processes cpu autoprofile process

0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
0x00000000 0x00000000
```

Related Commands

Command	Description
show processes cpu	Displays detailed CPU utilization statistics.

show processes cpu extended

To see an extended CPU load report, use the **show processes cpu extended** command in user EXEC or privileged EXEC mode.

show processes cpu extended [**history**]

Syntax Description

history	(Optional) Displays the extended CPU load statistics for the entire history available, as configured by the **process cpu extended** [**history** *history-size*] command. The absence of the **history** keyword displays only the last report.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show processes cpu extended** command:

```
Router# show processes cpu extended

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

```
                  avg/max
################################################################################
```

The following is sample output from the **show processes cpu extended history** command:

```
Router# show processes cpu extended history

################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:04:26
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           179          12          0/12                  2/9
Low               18           0          0/12                  1/3
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
-------------------------
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:04:21
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           174           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
------------------------
################################################################################
Global Statistics
-----------------
5 sec CPU util 0%/0% Timestamp 21:03:31
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           176          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
------------------------
```

Related Commands

Command	Description
process cpu extended	Collects the extended CPU load for the specified history size.

show resource all

To display the details of a Resource Owner (RO), use the **show resource all** command in user EXEC or privileged EXEC mode.

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

Syntax Description

brief	(Optional) Displays the brief details of the ROs.
detail	(Optional) Displays all the details of the ROs.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show resource all** command:

```
Router# show resource all

Resource Owner: cpu
Resource User Type: iosprocess
Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           0        13          0  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223        2872     36029         79  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225      352744     33446      10546  0.00%  0.20%  0.17% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
```

```
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         2          0  0.00%  0.00%  0.00% AAA high-capacit
  Resource User: Policy Manager(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777232           0         1          0  0.00%  0.00%  0.00% Policy Manager
  Resource User: Crash writer(ID: 0x1000011)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777233           0         1          0  0.00%  0.00%  0.00% Crash writer
  Resource User: RO Notify Timers(ID: 0x1000012)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777234           0         1          0  0.00%  0.00%  0.00% RO Notify Timers
  Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777235           0         1          0  0.00%  0.00%  0.00% RMI RM Notify Wa
  Resource User: EnvMon(ID: 0x1000014)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777236       11164     92859        120  0.00%  0.00%  0.00% EnvMon
  Resource User: IPC Dynamic Cache(ID: 0x1000015)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777237           0      3004          0  0.00%  0.00%  0.00% IPC Dynamic Cach
  Resource User: IPC Periodic Timer(ID: 0x1000017)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777239           0    180082          0  0.00%  0.00%  0.00% IPC Periodic Tim
  Resource User: IPC Managed Timer(ID: 0x1000018)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777240         572     79749          7  0.00%  0.00%  0.00% IPC Managed Time
  Resource User: IPC Deferred Port Closure(ID: 0x1000019)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777241           4    180088          0  0.00%  0.00%  0.00% IPC Deferred Por
  Resource User: IPC Seat Manager(ID: 0x100001A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777242       97560   1408799         69  0.23%  0.02%  0.00% IPC Seat Manager
  Resource User: IPC Session Service(ID: 0x100001B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777243           0         1          0  0.00%  0.00%  0.00% IPC Session Serv
  Resource User: ARP Input(ID: 0x100001C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777244          20      3082          6  0.00%  0.00%  0.00% ARP Input
  Resource User: EEM ED Syslog(ID: 0x100001D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Chunk Elements :
Allocated Size(b): 35152564 Count: 91901 Freed Size(b): 31793276 Count: 39159

Processor memory
Total Memory held : 46596832 bytes
```

```
pc = 0x403089D8, size =  10499724, count =    1
pc = 0x402996C8, size =   6737976, count = 8298
pc = 0x402F0C9C, size =   5821352, count =   10
pc = 0x40A25134, size =   4194324, count =    1
pc = 0x41D6D414, size =   1704144, count =   52
pc = 0x40451BE0, size =   1114180, count =   17
pc = 0x402D0DAC, size =    917600, count =    1
pc = 0x4043E5F4, size =    836076, count = 12291
pc = 0x404A276C, size =    617476, count =    1
pc = 0x41CDED1C, size =    569844, count =  125
pc = 0x4194C2D0, size =    524292, count =    1
pc = 0x405FD93C, size =    516100, count =    1
pc = 0x414D67AC, size =    473224, count =  199
pc = 0x41016294, size =    458756, count =    1
pc = 0x4046E618, size =    432096, count =    1
pc = 0x400A1134, size =    412420, count =    1
pc = 0x402ABB50, size =    317316, count =   93
pc = 0x41D53668, size =    262148, count =    1
pc = 0x4049BA04, size =    206640, count =   84
pc = 0x41E3FE30, size =    196620, count =    3
pc = 0x40B05214, size =    196612, count =    1
pc = 0x40494D94, size =    180180, count = 4095
pc = 0x402ABB6C, size =    144708, count =   93
pc = 0x41586A38, size =    144004, count =    1
pc = 0x4030B408, size =    140028, count =    7
pc = 0x415090EC, size =    131768, count =    4
pc = 0x41E37B94, size =    131088, count =    4
pc = 0x4195C348, size =    131076, count =    1
pc = 0x400A1194, size =    124420, count =    1
pc = 0x41503BC4, size =    122768, count =    1
pc = 0x404E888C, size =    114660, count = 4095
pc = 0x40494D50, size =    114660, count = 4095
pc = 0x404D99B0, size =    114660, count = 4095
pc = 0x4023F5B4, size =     98312, count =    2
pc = 0x41E45894, size =     97456, count =  626
pc = 0x41E2D4C4, size =     91584, count =   12
pc = 0x416D9768, size =     84004, count =    1
pc = 0x40452790, size =     84000, count = 3000
pc = 0x40322A74, size =     81948, count =    7
pc = 0x41D0FF4C, size =     81924, count =    1
pc = 0x40E9F7B0, size =     81364, count =    1
pc = 0x414FB1BC, size =     78740, count =    2
pc = 0x414D4A64, size =     72916, count =    2
pc = 0x40328770, size =     72144, count =   36
pc = 0x414FA938, size =     71592, count =    2
pc = 0x414EF938, size =     71096, count =    2
pc = 0x41947EEC, size =     65540, count =    1
pc = 0x41935B5C, size =     65540, count =    1
pc = 0x4193A348, size =     65540, count =    1
pc = 0x4193FF5C, size =     65540, count =    1
pc = 0x41D6E32C, size =     65540, count =    1
pc = 0x41DD534C, size =     65540, count =    1
pc = 0x414B5870, size =     65540, count =    1
pc = 0x4078521C, size =     65540, count =    1
.
.
.
I/O memory
Total Memory held : 9816224 bytes
pc = 0x4029983C, size =   9791584, count = 8290
pc = 0x403EC2A4, size =      8208, count =    1
pc = 0x403F8CD0, size =      8208, count =    1
pc = 0x403EC2E0, size =      4112, count =    1
pc = 0x403F8D0C, size =      4112, count =    1
```

```
Resource User: Scheduler(ID: 0x1000002)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 13052 bytes
pc = 0x4037BCC8, size =     12004, count =    1
pc = 0x40327110, size =      1048, count =   24

Resource User: Dead(ID: 0x1000003)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 447448 bytes
pc = 0x404A276C, size =    395636, count =    5
pc = 0x4043E5F4, size =     18676, count =  271
pc = 0x40494D94, size =      6888, count =   82
pc = 0x4044B9E4, size =      6672, count =    6
pc = 0x40C8BAB4, size =      5780, count =   34
pc = 0x404943DC, size =      2836, count =   82
pc = 0x40494D50, size =      2796, count =   82
pc = 0x4044DAF0, size =      2224, count =    2
pc = 0x40393168, size =      1772, count =    1
pc = 0x40FF2688, size =       728, count =    6
pc = 0x40CBC5A4, size =       400, count =    4
pc = 0x40455144, size =       320, count =   10
pc = 0x40C9A8D8, size =       288, count =    8
pc = 0x40CADE10, size =       260, count =    5
pc = 0x40B19484, size =       256, count =    2
pc = 0x4052BD2C, size =       208, count =    4
pc = 0x40CADE50, size =       188, count =    5
pc = 0x4044FBD8, size =       184, count =    1
pc = 0x40A9B2F0, size =       184, count =    1
pc = 0x40CBC45C, size =       160, count =    2
pc = 0x4038BF34, size =       144, count =    2
pc = 0x40529610, size =       136, count =    2
pc = 0x405CF034, size =       104, count =    1
pc = 0x414D67AC, size =       104, count =    1
pc = 0x4038BF68, size =        88, count =    2
pc = 0x4044F078, size =        84, count =    3
pc = 0x41555624, size =        84, count =    1
pc = 0x40685250, size =        76, count =    1
pc = 0x40481AD4, size =        68, count =    1
pc = 0x4044DB18, size =        56, count =    2
pc = 0x401B6960, size =        48, count =    1

Resource User: Interrupt(ID: 0x1000004)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 39652 Count: 1070

Processor memory
Total Memory held : 0 bytes

Resource User: Memory RO RU(ID: 0x1000005)
Chunk Elements :
Allocated Size(b): 12320 Count: 120 Freed Size(b): 10164 Count: 99

Processor memory
Total Memory held : 131080 bytes
pc = 0x40357C54, size =     65540, count =    1
pc = 0x40357D98, size =     65540, count =    1
```

```
Resource User: Chunk Manager(ID: 0x1000006)
Chunk Elements :
Allocated Size(b): 124 Count: 6 Freed Size(b): 48 Count: 3

Processor memory
Total Memory held : 9788 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x40332490, size =      3008, count =    2
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: Load Meter(ID: 0x1000007)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 44 Count: 1

Processor memory
Total Memory held : 3780 bytes
pc = 0x4037BCC8, size =      3004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: Check heaps(ID: 0x1000009)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 44 Count: 1

Processor memory
Total Memory held : 7236 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x41E2B0D0, size =       324, count =    1
pc = 0x403604BC, size =       140, count =    1
pc = 0x40351D2C, size =        76, count =    1
pc = 0x40351CF8, size =        56, count =    1

Resource User: Pool Manager(ID: 0x100000A)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: Buffer RO RU(ID: 0x100000B)
Chunk Elements :
Allocated Size(b): 4960 Count: 40 Freed Size(b): 4092 Count: 33

Processor memory
Total Memory held : 0 bytes

Resource User: Timers(ID: 0x100000C)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 44 Count: 1
.
.
.
Resource User: PF_Init Process(ID: 0x100004F)
Chunk Elements :
Allocated Size(b): 8104 Count: 126 Freed Size(b): 1400 Count: 29

Processor memory
Total Memory held : 31204 bytes
pc = 0x4027EF10, size =     21540, count =    5
```

```
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4044DAF0, size =      1112, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x4038BF68, size =       308, count =    7
pc = 0x4038BF34, size =       280, count =    7
pc = 0x403604BC, size =       280, count =    2
pc = 0x41E45ED0, size =       240, count =    5
pc = 0x401FB400, size =       236, count =    5
pc = 0x40529610, size =       136, count =    2
pc = 0x4047D560, size =       108, count =    2
pc = 0x4038C114, size =        88, count =    2
pc = 0x4044DB18, size =        72, count =    1
pc = 0x40211DCC, size =        56, count =    2
pc = 0x4038E038, size =        44, count =    1
pc = 0x40402C98, size =        32, count =    1
pc = 0x40455144, size =        32, count =    1

Resource User: PF_Split Sync Process(ID: 0x1000052)
Chunk Elements :
Allocated Size(b): 6092 Count: 87 Freed Size(b): 5644 Count: 81

Processor memory
Total Memory held : 10356 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4060364C, size =      1760, count =   10
pc = 0x41E45894, size =       960, count =    2
pc = 0x4060AE18, size =       856, count =   10
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: RPC pf-split-rp(ID: 0x1000053)
Chunk Elements :
Allocated Size(b): 1348 Count: 20 Freed Size(b): 1304 Count: 19

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: RPC idprom-MP(ID: 0x1000054)
Chunk Elements :
Allocated Size(b): 4708 Count: 68 Freed Size(b): 4664 Count: 67

Processor memory
Total Memory held : 16648 bytes
pc = 0x405023D4, size =      9732, count =   18
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1
pc = 0x405D000C, size =       136, count =    1

Resource User: Net Input(ID: 0x1000055)
Chunk Elements :
Allocated Size(b): 88 Count: 2 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1
```

```
Resource User: Compute load avgs(ID: 0x1000056)
Chunk Elements :
Allocated Size(b): 11948724 Count: 215941 Freed Size(b): 11948724 Count: 215941

Processor memory
Total Memory held : 10720 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x404FC9C0, size =      3940, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: RTTYS Process(ID: 0x1000057)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: BACK CHECK(ID: 0x1000059)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: chkpt message handler(ID: 0x100005A)
Chunk Elements :
Allocated Size(b): 156 Count: 2 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x4037BCC8, size =      6004, count =    1
pc = 0x4035E160, size =       636, count =    1
pc = 0x403604BC, size =       140, count =    1

Resource User: cpf_process_msg_holdq(ID: 0x100005B)
Chunk Elements :
Allocated Size(b): 152 Count: 3 Freed Size(b): 0 Count: 0
.
.
.
Resource Owner: Buffer
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Getbufs  Retbufs  Holding  RU Name
1367     31237    4294937426 Init

Resource User: Scheduler(ID: 0x1000002)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Scheduler

Resource User: Dead(ID: 0x1000003)
Getbufs  Retbufs  Holding  RU Name
6        3        3        Dead

Resource User: Interrupt(ID: 0x1000004)
Getbufs  Retbufs  Holding  RU Name
221580   221580   0        Interrupt
```

```
Resource User: Memory RO RU(ID: 0x1000005)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Memory RO RU

Resource User: Chunk Manager(ID: 0x1000006)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Chunk Manager

Resource User: Load Meter(ID: 0x1000007)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Load Meter

Resource User: Check heaps(ID: 0x1000009)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Check heaps

Resource User: Pool Manager(ID: 0x100000A)
Getbufs  Retbufs  Holding  RU Name
5554     0        5554     Pool Manager

Resource User: Buffer RO RU(ID: 0x100000B)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Buffer RO RU

Resource User: Timers(ID: 0x100000C)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Timers

Resource User: Serial Background(ID: 0x100000D)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Serial Backgroun

Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
Getbufs  Retbufs  Holding  RU Name
0        0        0        AAA_SERVER_DEADT

Resource User: AAA high-capacity counters(ID: 0x100000F)
Getbufs  Retbufs  Holding  RU Name
0        0        0        AAA high-capacit

Resource User: Policy Manager(ID: 0x1000010)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Policy Manager

Resource User: Crash writer(ID: 0x1000011)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Crash writer

Resource User: RO Notify Timers(ID: 0x1000012)
Getbufs  Retbufs  Holding  RU Name
0        0        0        RO Notify Timers

Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
Getbufs  Retbufs  Holding  RU Name
0        0        0        RMI RM Notify Wa

.
.
.
Resource User: DHCPD Timer(ID: 0x100011B)
Getbufs  Retbufs  Holding  RU Name
0        0        0        DHCPD Timer

Resource User: DHCPD Database(ID: 0x100011C)
```

```
Getbufs  Retbufs  Holding  RU Name
0        0        0        DHCPD Database

Resource User: draco-oir-process:slot 2(ID: 0x100011E)
Getbufs  Retbufs  Holding  RU Name
0        0        0        draco-oir-proces

Resource User: SCP async: Draco-LC4(ID: 0x1000125)
Getbufs  Retbufs  Holding  RU Name
35849    243101   4294760044 SCP async: Draco

Resource User: IFCOM Msg Hdlr(ID: 0x1000127)
Getbufs  Retbufs  Holding  RU Name
2        2        0        IFCOM Msg Hdlr

Resource User: IFCOM Msg Hdlr(ID: 0x1000128)
Getbufs  Retbufs  Holding  RU Name
28       28       0        IFCOM Msg Hdlr

Resource User: Exec(ID: 0x100012C)
Getbufs  Retbufs  Holding  RU Name
912      912      0        Exec

Resource Owner: test_mem
Resource User Type: test_process
Resource User Type: mem_rut
Resource Owner: test_cpu
Resource User Type: test_process
Resource User Type: cpu_rut
```

The following is a sample output from the **show resource all brief** command:

```
Router# show resource all brief

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           0        13          0  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223        2872     36069         79  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225      353092     33481      10546  0.00%  0.17%  0.17% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
```

```
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         2          0  0.00%  0.00%  0.00% AAA high-capacit
  Resource User: Policy Manager(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777232           0         1          0  0.00%  0.00%  0.00% Policy Manager
  Resource User: Crash writer(ID: 0x1000011)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777233           0         1          0  0.00%  0.00%  0.00% Crash writer
  Resource User: RO Notify Timers(ID: 0x1000012)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777234           0         1          0  0.00%  0.00%  0.00% RO Notify Timers
  Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777235           0         1          0  0.00%  0.00%  0.00% RMI RM Notify Wa
  Resource User: EnvMon(ID: 0x1000014)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777236       11176     92958        120  0.00%  0.00%  0.00% EnvMon
  Resource User: IPC Dynamic Cache(ID: 0x1000015)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777237           0      3007          0  0.00%  0.00%  0.00% IPC Dynamic Cach
  Resource User: IPC Periodic Timer(ID: 0x1000017)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777239           0    180279          0  0.00%  0.00%  0.00% IPC Periodic Tim
  Resource User: IPC Managed Timer(ID: 0x1000018)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777240         572     79833          7  0.00%  0.00%  0.00% IPC Managed Time
  Resource User: IPC Deferred Port Closure(ID: 0x1000019)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777241           4    180285          0  0.00%  0.00%  0.00% IPC Deferred Por
  Resource User: IPC Seat Manager(ID: 0x100001A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777242       97684   1410183         69  0.00%  0.03%  0.00% IPC Seat Manager
  Resource User: IPC Session Service(ID: 0x100001B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777243           0         1          0  0.00%  0.00%  0.00% IPC Session Serv
  Resource User: ARP Input(ID: 0x100001C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777244          20      3085          6  0.00%  0.00%  0.00% ARP Input
  Resource User: EEM ED Syslog(ID: 0x100001D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777245           0        49          0  0.00%  0.00%  0.00% EEM ED Syslog
  Resource User: DDR Timers(ID: 0x100001E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777246           0         2          0  0.00%  0.00%  0.00% DDR Timers
  Resource User: Dialer event(ID: 0x100001F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777247           0         2          0  0.00%  0.00%  0.00% Dialer event
  Resource User: Entity MIB API(ID: 0x1000020)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777248          28        16       1750  0.00%  0.00%  0.00% Entity MIB API
  Resource User: Compute SRP rates(ID: 0x1000021)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
```

```
16777249           0     18037          0  0.00%  0.00%  0.00% Compute SRP rate
  Resource User: SERIAL A'detect(ID: 0x1000022)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777250           0         1          0  0.00%  0.00%  0.00% SERIAL A'detect
  Resource User: GraphIt(ID: 0x1000023)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777251           0    180267          0  0.00%  0.00%  0.00% GraphIt
  Resource User: rf proxy rp agent(ID: 0x1000024)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777252          40       416         96  0.00%  0.00%  0.00% rf proxy rp agen
  Resource User: HC Counter Timers(ID: 0x1000025)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777253          60     41360          1  0.00%  0.00%  0.00% HC Counter Timer
  Resource User: Snmp ICC Process(ID: 0x1000026)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777254           0         1          0  0.00%  0.00%  0.00% Snmp ICC Process
  Resource User: Cat6k SNMP(ID: 0x1000027)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777255          20        29        689  0.00%  0.00%  0.00% Cat6k SNMP
  Resource User: Cat6k SNMP Trap handler(ID: 0x1000028)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777256           0         7          0  0.00%  0.00%  0.00% Cat6k SNMP Trap
  Resource User: Critical Bkgnd(ID: 0x1000029)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777257           0         1          0  0.00%  0.00%  0.00% Critical Bkgnd
  Resource User: Net Background(ID: 0x100002A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777258         112     44787          2  0.00%  0.00%  0.00% Net Background
  Resource User: Logger(ID: 0x100002B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777259           0        50          0  0.00%  0.00%  0.00% Logger
  Resource User: TTY Background(ID: 0x100002C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777260           0    180263          0  0.00%  0.00%  0.00% TTY Background
  Resource User: Per-Second Jobs(ID: 0x100002D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777261          52    180549          0  0.00%  0.00%  0.00% Per-Second Jobs
  Resource User: Per-minute Jobs(ID: 0x100002E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource User: Exec(ID: 0x100012C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777516        8964       965       9289  0.39%  0.66%  1.55% Exec
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Processor memory
Allocated   Freed  Holding   Blocks
55233064  8636232 46596832    48832

I/O memory
Allocated   Freed  Holding   Blocks
 9816224        0  9816224     8294

  Resource User: Scheduler(ID: 0x1000002)
Processor memory
Allocated   Freed  Holding   Blocks
   13052        0    13052       25

  Resource User: Dead(ID: 0x1000003)
Processor memory
Allocated   Freed  Holding   Blocks
```

```
  687916   240468   447448      630

  Resource User: Interrupt(ID: 0x1000004)
Processor memory
Allocated   Freed  Holding   Blocks
       0       0        0        0

  Resource User: Memory RO RU(ID: 0x1000005)
Processor memory
Allocated   Freed  Holding   Blocks
  131080       0   131080        2

  Resource User: Chunk Manager(ID: 0x1000006)
Processor memory
Allocated   Freed  Holding   Blocks
   14300    4512     9788        5

  Resource User: Load Meter(ID: 0x1000007)
Processor memory
Allocated   Freed  Holding   Blocks
    3920     140     3780        3

  Resource User: Check heaps(ID: 0x1000009)
Processor memory
Allocated   Freed  Holding   Blocks
    7376     140     7236        6

  Resource User: Pool Manager(ID: 0x100000A)
Processor memory
Allocated   Freed  Holding   Blocks
    6780       0     6780        3

  Resource User: Buffer RO RU(ID: 0x100000B)
Processor memory
Allocated   Freed  Holding   Blocks
       0       0        0        0

  Resource User: Timers(ID: 0x100000C)
Processor memory
Allocated   Freed  Holding   Blocks
    6920     140     6780        3

  Resource User: Serial Background(ID: 0x100000D)
Processor memory
Allocated   Freed  Holding   Blocks
    6920     140     6780        3
.
.
.
Resource User: IFCOM Msg Hdlr(ID: 0x1000128)
Getbufs  Retbufs  Holding  RU Name
28       28       0        IFCOM Msg Hdlr

  Resource User: Exec(ID: 0x100012C)
Getbufs  Retbufs  Holding  RU Name
1404     1404     0        Exec

Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
```

The following is sample output from the **show resource all detailed** command:

```
Router# show resource all detailed

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
  Resource User: Scheduler(ID: 0x1000002)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
  Resource User: Dead(ID: 0x1000003)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
  Resource User: Interrupt(ID: 0x1000004)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
  Resource User: Memory RO RU(ID: 0x1000005)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
  Resource User: Chunk Manager(ID: 0x1000006)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           0        13          0  0.00%  0.00%  0.00% Chunk Manager
  Resource User: Load Meter(ID: 0x1000007)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223        2872     36075         79  0.00%  0.00%  0.00% Load Meter
  Resource User: Check heaps(ID: 0x1000009)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225      353168     33486      10546  0.00%  0.10%  0.15% Check heaps
  Resource User: Pool Manager(ID: 0x100000A)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
  Resource User: Buffer RO RU(ID: 0x100000B)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
  Resource User: Timers(ID: 0x100000C)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
  Resource User: Serial Background(ID: 0x100000D)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777229           0         2          0  0.00%  0.00%  0.00% Serial Backgroun
  Resource User: AAA_SERVER_DEADTIME(ID: 0x100000E)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777230           0         1          0  0.00%  0.00%  0.00% AAA_SERVER_DEADT
  Resource User: AAA high-capacity counters(ID: 0x100000F)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777231           0         2          0  0.00%  0.00%  0.00% AAA high-capacit
  Resource User: Policy Manager(ID: 0x1000010)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777232           0         1          0  0.00%  0.00%  0.00% Policy Manager
  Resource User: Crash writer(ID: 0x1000011)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777233           0         1          0  0.00%  0.00%  0.00% Crash writer
  Resource User: RO Notify Timers(ID: 0x1000012)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777234           0         1          0  0.00%  0.00%  0.00% RO Notify Timers
  Resource User: RMI RM Notify Watched Policy(ID: 0x1000013)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777235           0         1          0  0.00%  0.00%  0.00% RMI RM Notify Wa
  Resource User: EnvMon(ID: 0x1000014)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777236       11176     92958        120  0.00%  0.00%  0.00% EnvMon
  Resource User: IPC Dynamic Cache(ID: 0x1000015)
```

```
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777237           0      3008          0  0.00%  0.00%  0.00% IPC Dynamic Cach
  Resource User: IPC Periodic Timer(ID: 0x1000017)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Chunk Elements :
Allocated Size(b): 35152564 Count: 91901 Freed Size(b): 31793276 Count: 39159

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
4393BAA0 0010499772 00000000 4433F15C 001 513DD000  *Init*
4433F15C 0000012852 4393BAA0 44342390 001 513DD000  *Init*
44342390 0000005052 4433F15C 4434374C 001 513DD000  List Headers
4434374C 0000000096 44342390 443437AC 001 513DD000  *Init*
443437AC 0000000096 4434374C 4434380C 001 513DD000  *Init*
4434380C 0000000096 443437AC 4434386C 001 513DD000  *Init*
4434386C 0000000096 4434380C 443438CC 001 513DD000  *Init*
443438CC 0000000096 4434386C 4434392C 001 513DD000  *Init*
4434392C 0000004356 443438CC 44344A30 001 513DD000  TTY data
44344A30 0000000564 4434392C 44344C64 001 513DD000  TTY Output Buf
44344C64 0000000096 44344A30 44344CC4 001 513DD000  *Init*
44344CC4 0000001552 44344C64 443452D4 001 513DD000  Watched messages
443452D4 0000010052 44344CC4 44347A18 001 513DD000  Watched Boolean
44347A18 0000001552 443452D4 44348028 001 513DD000  Watched Semaphore
44348028 0000000380 44347A18 443481A4 001 513DD000  Watched Message Queue
443481A4 0000003052 44348028 44348D90 001 513DD000  Read/Write Locks
44348D90 0000020052 443481A4 4434DBE4 001 513DD000  RMI-RO_RU Chunks
4434DBE4 0000000116 44348D90 4434DC58 001 513DD000  Resource Owner IDs
4434DC58 0000001552 4434DBE4 4434E268 001 513DD000  String-DB entries
4434E268 0000000532 4434DC58 4434E47C 001 513DD000  String-DB handles
4434E47C 0000000076 4434E268 4434E4C8 001 513DD000  NameDB String
4434E4C8 0000000116 4434E47C 4434E53C 001 513DD000  Resource User Type IDs
4434E53C 0000000184 4434E4C8 4434E5F4 001 513DD000  *Init*
4434E5F4 0000002100 4434E53C 4434EE28 001 513DD000  Resource Owner IDs
4434EE28 0000000076 4434E5F4 4434EE74 001 513DD000  NameDB String
4434EE74 0000000076 4434EE28 4434EEC0 001 513DD000  NameDB String
4434EEC0 0000065588 4434EE74 4435EEF4 001 513DD000  Buffer RU Notify Chunks
44360754 0000000076 44360698 443607A0 001 513DD000  *Init*
443607A0 0000002100 44360754 44360FD4 001 513DD000  Resource User Type IDs
44360FD4 0000004148 443607A0 44362008 001 513DD000  Resource User IDs
44362008 0000000076 44360FD4 44362054 001 513DD000  NameDB String
44362054 0000000076 44362008 443620A0 001 513DD000  NameDB String
443620A0 0000000096 44362054 44362100 001 513DD000  *Init*
443623AC 0000000076 44362100 443623F8 001 513DD000  NameDB String
443623F8 0000010052 443623AC 44364B3C 001 513DD000  List Elements
44364B3C 0000010052 443623F8 44367280 001 513DD000  List Elements
4436758C 0000001552 4436752C 44367B9C 001 513DD000  Reg Function iList
44367B9C 0000000164 4436758C 44367C40 001 513DD000  *Init*
44367C40 0000000076 44367B9C 44367C8C 001 513DD000  Parser Linkage
44367C8C 0000000076 44367C40 44367CD8 001 513DD000  Parser Linkage
44367CD8 0000000076 44367C8C 44367D24 001 513DD000  Parser Linkage
44367D70 0000000076 44367D24 44367DBC 001 513DD000  Parser Linkage
44367DBC 0000000076 44367D70 44367E08 001 513DD000  Cond Debug definition
44367E08 0000000076 44367DBC 44367E54 001 513DD000  Parser Linkage
44367E54 0000000076 44367E08 44367EA0 001 513DD000  Cond Debug definition
44367EA0 0000000076 44367E54 44367EEC 001 513DD000  Cond Debug definition
44367EEC 0000000076 44367EA0 44367F38 001 513DD000  Cond Debug definition
44367F38 0000000076 44367EEC 44367F84 001 513DD000  Cond Debug definition
44367F84 0000000384 44367F38 44368104 001 513DD000  *Init*
```

```
4436B5C8 0000000076 4436B57C 4436B614 001 513DD000  Init
4436B614 0000000076 4436B5C8 4436B660 001 513DD000  Init
4436B660 0000000076 4436B614 4436B6AC 001 513DD000  Init
4436BC04 0000000076 4436BBB8 4436BC50 001 513DD000  Init
4436BC50 0000003460 4436BC04 4436C9D4 001 513DD000  *Hardware IDB*
4436C9D4 0000000076 4436BC50 4436CA20 001 513DD000  Init
4436CA20 0000001080 4436C9D4 4436CE58 001 513DD000  Index Table Block
4436CE58 0000000076 4436CA20 4436CEA4 001 513DD000  Init
4436CEA4 0000000076 4436CE58 4436CEF0 001 513DD000  Init
4436CEF0 0000000308 4436CEA4 4436D024 001 513DD000  Init
4436D024 0000000076 4436CEF0 4436D070 001 513DD000  NameDB String
4436D070 0000000104 4436D024 4436D0D8 001 513DD000  NameDB String
4436D434 0000000096 4436D188 4436D494 001 513DD000  Init
4436D740 0000000096 4436D494 4436D7A0 001 513DD000  Init
4436D7A0 0000010052 4436D740 4436FEE4 001 513DD000  Packet Elements
4436FEE4 0000000372 4436D7A0 44370058 001 513DD000  Pool Info
44370058 0000000372 4436FEE4 443701CC 001 513DD000  Pool Info
443701CC 0000000372 44370058 44370340 001 513DD000  Pool Info
44370340 0000000860 443701CC 4437069C 001 513DD000  *Packet Header*
4437069C 0000000372 44370340 44370810 001 513DD000  Pool Info
44370810 0000000860 4437069C 44370B6C 001 513DD000  *Packet Header*
44370B6C 0000000860 44370810 44370EC8 001 513DD000  *Packet Header*
44370EC8 0000000860 44370B6C 44371224 001 513DD000  *Packet Header*
44371224 0000000860 44370EC8 44371580 001 513DD000  *Packet Header*
44371580 0000000860 44371224 443718DC 001 513DD000  *Packet Header*
443718DC 0000000860 44371580 44371C38 001 513DD000  *Packet Header*
44371C38 0000000860 443718DC 44371F94 001 513DD000  *Packet Header*
44371F94 0000000860 44371C38 443722F0 001 513DD000  *Packet Header*
443722F0 0000000860 44371F94 4437264C 001 513DD000  *Packet Header*
4437264C 0000000860 443722F0 443729A8 001 513DD000  *Packet Header*
443729A8 0000000860 4437264C 44372D04 001 513DD000  *Packet Header*
.
.
.
Resource User: Compute SRP rates(ID: 0x1000021)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
446D502C 0000006052 446D4D5C 446D67D0 001 513DD000  Init
446D67D0 0000000188 446D502C 446D688C 001 513DD000  Process Events
5055163C 0000000684 505512CC 505518E8 001 513DD000  Init
  Resource User: SERIAL A'detect(ID: 0x1000022)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 0 Count: 0

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
44722FCC 0000000684 4471DE58 44723278 001 513DD000  Init
50598A4C 0000006052 505989E8 5059A1F0 001 513DD000  Init
5059A1F0 0000000188 50598A4C 5059A2AC 001 513DD000  Process Events
  Resource User: GraphIt(ID: 0x1000023)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 44 Count: 1

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
447235B8 0000000684 4472356C 44723864 001 513DD000  Init
5059A8A8 0000006052 5059A350 5059C04C 001 513DD000  Init
5059C04C 0000000188 5059A8A8 5059C108 001 513DD000  Process Events
  Resource User: rf proxy rp agent(ID: 0x1000024)
Chunk Elements :
Allocated Size(b): 39056 Count: 504 Freed Size(b): 33756 Count: 452
```

```
Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
446B752C 0000000144 446B74D4 446B75BC 001 513DD000  NameDB String
44728FC0 0000000684 44728F74 4472926C 001 513DD000  Init
44B19780 0000001160 44B1867C 44B19C08 001 513DD000  IPC Port
44B204A0 0000000148 44B2042C 44B20534 001 513DD000  IPC Name String
44B220E8 0000000096 44B2202C 44B22148 001 513DD000  rf proxy rp agent
44B22148 0000001160 44B220E8 44B225D0 001 513DD000  IPC Port
44B22938 0000000076 44B2287C 44B22984 001 513DD000  NameDB String
44B22984 0000000096 44B22938 44B229E4 001 513DD000  rf proxy rp agent
44B22D4C 0000000076 44B22C90 44B22D98 001 513DD000  NameDB String
44B22D98 0000000096 44B22D4C 44B22DF8 001 513DD000  rf proxy rp agent
44B23160 0000000076 44B230A4 44B231AC 001 513DD000  NameDB String
44B231AC 0000000096 44B23160 44B2320C 001 513DD000  rf proxy rp agent
44B2320C 0000000076 44B231AC 44B23258 001 513DD000  IPC Name String
50543ABC 0000000104 50543A00 50543B24 001 513DD000  IPC Name
5061CC34 0000000188 5059EC00 5061CCF0 001 513DD000  Process Events
5061CDB4 0000006052 5061CD68 5061E558 001 513DD000  Init
50A8780C 0000000132 50A877C0 50A87890 001 513DD000  IPC Name String
50AC8094 0000065588 50AC7C0C 50AD80C8 001 513DD000  EvtMgr active chunk
50AD986C 0000001160 50AD80C8 50AD9CF4 001 513DD000  IPC Port
  Resource User: HC Counter Timers(ID: 0x1000025)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0
.
.
.
Resource User: NetFlow Agg Task(ID: 0x1000114)
Getbufs  Retbufs  Holding  RU Name
0        0        0        NetFlow Agg Task

  Resource User: CWAN OIR IPC Ready Process(ID: 0x1000115)
Getbufs  Retbufs  Holding  RU Name
0        0        0        CWAN OIR IPC Rea

  Resource User: PF Clock Process(ID: 0x1000116)
Getbufs  Retbufs  Holding  RU Name
0        0        0        PF Clock Process

  Resource User: CEF IPC Background(ID: 0x1000117)
Getbufs  Retbufs  Holding  RU Name
0        0        0        CEF IPC Backgrou

  Resource User: RTTYS Process(ID: 0x1000118)
Getbufs  Retbufs  Holding  RU Name
0        0        0        RTTYS Process

  Resource User: DHCPD Timer(ID: 0x100011B)
Getbufs  Retbufs  Holding  RU Name
0        0        0        DHCPD Timer

  Resource User: DHCPD Database(ID: 0x100011C)
Getbufs  Retbufs  Holding  RU Name
0        0        0        DHCPD Database

  Resource User: draco-oir-process:slot 2(ID: 0x100011E)
Getbufs  Retbufs  Holding  RU Name
0        0        0        draco-oir-proces

  Resource User: SCP async: Draco-LC4(ID: 0x1000125)
Getbufs  Retbufs  Holding  RU Name
35908    243517   4294759687 SCP async: Draco
```

```
  Resource User: IFCOM Msg Hdlr(ID: 0x1000127)
Getbufs  Retbufs  Holding  RU Name
2        2        0        IFCOM Msg Hdlr

  Resource User: IFCOM Msg Hdlr(ID: 0x1000128)
Getbufs  Retbufs  Holding  RU Name
28       28       0        IFCOM Msg Hdlr

  Resource User: Exec(ID: 0x100012C)
Getbufs  Retbufs  Holding  RU Name
17552    17552    0        Exec

Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
```

Table 88 describes the significant fields shown in the display.

Table 88 show resource all Field Descriptions

Field	Description
Runtime(ms)	The runtime of the process in milliseconds.
Invoked	The number of times a Resource User (RU) has been allowed to run.
uSecs	The amount of runtime per invocation in microseconds.
Allocated Size(b)	The number of bytes of memory that is allocated.
Freed Size(b)	The number of bytes of memory that is freed.
Count	The number of elements that are allocated or freed. For example, if two elements of 50 bytes each are allocated, then the allocated count is 2 and allocated size is 100.
pc	Displays the details of the memory that is held by a process. Each line of the output displays one or more blocks of memory. The pc is the allocator pc of a particular block of memory.
size	The total size of memory allocated to each block. The sum of the size of all blocks is equivalent to the total memory held by the process.
count	The count is the number of blocks of memory.
Getbufs	The number of buffers allocated by the RU.
Retbufs	The number of buffers freed by the RU.
Holding	The number of buffers the RU is holding currently.

Related Commands

Command	Description
buffer public	Enters the buffer owner configuration mode and sets thresholds for buffer usage.
cpu interrupt	Enters the CPU owner configuration mode and sets thresholds for interrupt level CPU utilization.

Command	Description
cpu process	Enters the CPU owner configuration mode and sets thresholds for processor level CPU utilization.
cpu total	Enters the CPU owner configuration mode and sets thresholds for total CPU utilization.
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
memory io	Enters the memory owner configuration mode and sets threshold values for I/O memory.
memory processor	Enters the memory owner configuration mode and sets threshold values for processor memory.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show resource database	Displays the database details of ROs.
show resource owner	Displays the RO details.
show resource relationship	Displays the relationship between the RUs and the ROs.

show resource database

To display the details of a resource owner, use the **show resource database** command in user EXEC or privileged EXEC mode.

show resource database

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show resource database** command:

```
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.
Owner: test_RO4                 Id:0xB
Owner's list of monitors is empty.
Owner: test_RO5                 Id:0xC
Owner's list of monitors is empty.
.
.
.
List of all Resource Usertypes :
RUT: iosprocess               Id:0x1
RUT: test_process             Id:0x2
RUT: mem_rut                  Id:0x3
RUT: cpu_rut                  Id:0x4
```

```
RUT: test_RUT0                Id:0x5
RUT: test_RUT1                Id:0x6
RUT: test_RUT2                Id:0x7
RUT: test_RUT3                Id:0x8
RUT: test_RUT4                Id:0x9
RUT: test_RUT5                Id:0xA
.
.
.
List of all Resource User Groups :

List of all Resource Users :
usertype: iosprocess               Id:0x1
 user: Init                      Id:0x1000001, priority:0
 user: Scheduler                 Id:0x1000002, priority:0
 user: Dead                      Id:0x1000003, priority:0
 user: Interrupt                 Id:0x1000004, priority:0
 user: Memory RO RU              Id:0x1000005, priority:0
 user: Chunk Manager             Id:0x1000006, priority:1
 user: Load Meter                Id:0x1000007, priority:1
 user: Check heaps               Id:0x1000009, priority:4
 user: Pool Manager              Id:0x100000A, priority:1
 user: Buffer RO RU              Id:0x100000B, priority:0
 user: Timers                    Id:0x100000C, priority:3
 user: Serial Background         Id:0x100000D, priority:3
 user: ALARM_TRIGGER_SCAN        Id:0x100000E, priority:4
 user: AAA_SERVER_DEADTIME       Id:0x100000F, priority:4
 user: AAA high-capacity counter Id:0x1000010, priority:3
 user: Policy Manager            Id:0x1000011, priority:3
 user: Crash writer              Id:0x1000012, priority:3
 user: RO Notify Timers          Id:0x1000013, priority:3
 user: RMI RM Notify Watched Pol Id:0x1000014, priority:3
 user: EnvMon                    Id:0x1000015, priority:3
 user: OIR Handler               Id:0x1000016, priority:3
 user: IPC Dynamic Cache         Id:0x1000017, priority:3
 user: IPC Zone Manager          Id:0x1000018, priority:3
 user: IPC Periodic Timer        Id:0x1000019, priority:3
 user: IPC Managed Timer         Id:0x100001A, priority:3
 user: IPC Deferred Port Closure Id:0x100001B, priority:3
.
.
.
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.
```

Related Commands

Command	Description
buffer public	Enters the buffer owner configuration mode and sets thresholds for buffer usage.
cpu interrupt	Enters the CPU owner configuration mode and sets thresholds for interrupt level CPU utilization.
cpu process	Enters the CPU owner configuration mode and sets thresholds for processor level CPU utilization.

Command	Description
cpu total	Enters the CPU owner configuration mode and sets thresholds for total CPU utilization.
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
memory io	Enters the memory owner configuration mode and sets threshold values for I/O memory.
memory processor	Enters the memory owner configuration mode and sets threshold values for processor memory.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show resource all	Displays all the resource details.
show resource owner	Displays the RO details.
show resource relationship	Displays the relationship between the RUs and the ROs.

show resource owner

To display the details of a resource owner (RO), use the **show resource owner** command in user EXEC or privileged EXEC mode.

show resource owner {*resource-owner-name* | **all**} **user** {*resource-user-type-name* | **all**} [**brief** [**triggers**] | **detailed** [**triggers**] | **triggers**]

Syntax Description

resource-owner-name	Name of the specified RO whose details are displayed.
all	Displays details of all the ROs.
user	Displays details of the specified resource user (RU) type.
resource-user-type-name	Single resource user type.
all	Displays details of all the resource user types.
brief	(Optional) Displays brief details.
detailed	(Optional) Displays complete details.
triggers	(Optional) Displays the triggers.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show resource owner** command:

```
Router# show resource owner all user all

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
Resource User: Scheduler(ID: 0x1000002)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource User: Dead(ID: 0x1000003)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource User: Interrupt(ID: 0x1000004)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource User: Memory RO RU(ID: 0x1000005)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource User: Chunk Manager(ID: 0x1000006)
```

```
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
Resource User: Load Meter(ID: 0x1000007)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223           4       292         13  0.00%  0.00%  0.00% Load Meter
Resource User: Check heaps(ID: 0x1000009)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225         376       192       1958  0.00%  0.02%  0.00% Check heaps
Resource User: Pool Manager(ID: 0x100000A)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
Resource User: Buffer RO RU(ID: 0x100000B)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
Resource User: Timers(ID: 0x100000C)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
Resource User: Serial Background(ID: 0x100000D)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
.
.
.
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Chunk Elements :
Allocated Size(b): 25967632 Count: 46612 Freed Size(b): 21487684 Count: 26053

Processor memory
Total Memory held : 15250376 bytes
pc = 0x6072D840, size =   4040536, count =    6
pc = 0x6034E040, size =   1937508, count =    2
pc = 0x6070DAF0, size =    560096, count =    1
pc = 0x606D7530, size =    556220, count =  685
pc = 0x613AFA74, size =    350972, count =   25
pc = 0x60ECA4F0, size =    280004, count =    1
pc = 0x606DEC1C, size =    270600, count =  100
pc = 0x616EF268, size =    262148, count =    1
pc = 0x6085C318, size =    196620, count =    3
pc = 0x61479630, size =    144004, count =    1
pc = 0x613E1DB0, size =    131768, count =    4
.
.
.
I/O memory
Total Memory held : 4059856 bytes
pc = 0x606DEC30, size =   3408704, count =   52
pc = 0x606DEB94, size =    442464, count =    6
pc = 0x606D76A4, size =    179872, count =  146
pc = 0x600ED530, size =     16448, count =    4
pc = 0x600ED498, size =      8256, count =    4
pc = 0x6080D3F0, size =      4112, count =    1

  Resource User: Scheduler(ID: 0x1000002)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 12172 bytes
pc = 0x607B44F0, size =     12004, count =    1
pc = 0x607643B8, size =       168, count =    4

.
.
```

```
.
Resource User: Critical Bkgnd(ID: 0x1000026)
Chunk Elements :
Allocated Size(b): 44 Count: 1 Freed Size(b): 0 Count: 0

Processor memory
Total Memory held : 6780 bytes
pc = 0x607B44F0, size =      6004, count =    1
pc = 0x6079CB28, size =       636, count =    1
pc = 0x6079EE84, size =       140, count =    1
.
.
.
Resource Owner: Buffer
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Getbufs  Retbufs  Holding  RU Name
319      51       268      Init

Resource User: Scheduler(ID: 0x1000002)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Scheduler

Resource User: Dead(ID: 0x1000003)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Dead

Resource User: Interrupt(ID: 0x1000004)
Getbufs  Retbufs  Holding  RU Name
1356     1356     0        Interrupt

Resource User: Memory RO RU(ID: 0x1000005)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Memory RO RU

Resource User: Chunk Manager(ID: 0x1000006)
Getbufs  Retbufs  Holding  RU Name
0        0        0        Chunk Manager

.
.
.
Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
Resource User: test_RU0(ID: 0x4000001)
>>>RU: Blank
Resource User: test_RU1(ID: 0x4000002)
>>>RU: Blank
Resource User: test_RU2(ID: 0x4000003)
>>>RU: Blank
Resource User: test_RU3(ID: 0x4000004)
>>>RU: Blank
.
.
.
Resource User Type: test_RUT143
 Resource User Type: test_RUT144
 Resource User Type: test_RUT145
 Resource User Type: test_RUT146
 Resource User Type: test_RUT147
```

The following is sample output from the **show resource owner all user all brief** command:

```
Router# show resource owner all user all brief

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
Resource User: Scheduler(ID: 0x1000002)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource User: Dead(ID: 0x1000003)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource User: Interrupt(ID: 0x1000004)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource User: Memory RO RU(ID: 0x1000005)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource User: Chunk Manager(ID: 0x1000006)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
Resource User: Load Meter(ID: 0x1000007)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223           4       322         12  0.00%  0.01%  0.00% Load Meter
Resource User: Check heaps(ID: 0x1000009)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225         424       214       1981  0.00%  0.04%  0.00% Check heaps
.
.
.
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Processor memory
Allocated   Freed  Holding   Blocks
21916780  6666404 15250376     8688

I/O memory
Allocated   Freed  Holding   Blocks
 4059856        0  4059856      213

Resource User: Scheduler(ID: 0x1000002)
Processor memory
Allocated   Freed  Holding   Blocks
   12172        0    12172        5
.
.
.
Resource Owner: test_mem
 Resource User Type: test_process
 Resource User Type: mem_rut
Resource Owner: test_cpu
 Resource User Type: test_process
 Resource User Type: cpu_rut
Resource User: test_RU0(ID: 0x4000001)
>>>RU: Blank
Resource User: test_RU1(ID: 0x4000002)
>>>RU: Blank
Resource User: test_RU2(ID: 0x4000003)
>>>RU: Blank
```

```
Resource User: test_RU3(ID: 0x4000004)
>>>RU: Blank
Resource User: test_RU4(ID: 0x4000005)
>>>RU: Blank
.
.
.
Resource Owner: test_RO0
Resource User Type: test_RUT0
Resource User Type: test_RUT1
Resource User Type: test_RUT2
Resource User Type: test_RUT3
Resource User Type: test_RUT4
Resource User Type: test_RUT5
Resource User Type: test_RUT6
Resource User Type: test_RUT7
Resource User Type: test_RUT8
Resource User Type: test_RUT9
Resource User Type: test_RUT10
Resource User Type: test_RUT11
Resource User Type: test_RUT12
Resource User Type: test_RUT13
Resource User Type: test_RUT14
Resource User Type: test_RUT15
Resource User Type: test_RUT16
```

The following is sample output from the **show resource owner all user all brief triggers** command:

```
Router# show resource owner all user all brief triggers

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
Resource User: Scheduler(ID: 0x1000002)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource User: Dead(ID: 0x1000003)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource User: Interrupt(ID: 0x1000004)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource User: Memory RO RU(ID: 0x1000005)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource User: Chunk Manager(ID: 0x1000006)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
.
.
.
Resource Owner: test_mem
Resource User Type: test_process
Resource User Type: mem_rut
Resource Owner: test_cpu
Resource User Type: test_process
Resource User Type: cpu_rut
Resource User: test_RU0(ID: 0x4000001)
>>>RU: Blank
Resource User: test_RU1(ID: 0x4000002)
>>>RU: Blank
Resource User: test_RU2(ID: 0x4000003)
```

```
>>>RU: Blank
Resource User: test_RU3(ID: 0x4000004)
>>>RU: Blank
Resource User: test_RU4(ID: 0x4000005)
>>>RU: Blank
Resource User: test_RU5(ID: 0x4000006)
>>>RU: Blank
```

The following is sample output from the **show resource owner all user all detailed** command:

```
Router# show resource owner all user all detailed

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
Resource User: Scheduler(ID: 0x1000002)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource User: Dead(ID: 0x1000003)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource User: Interrupt(ID: 0x1000004)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource User: Memory RO RU(ID: 0x1000005)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource User: Chunk Manager(ID: 0x1000006)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777222           4         3       1333  0.00%  0.00%  0.00% Chunk Manager
Resource User: Load Meter(ID: 0x1000007)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777223           4       353         11  0.00%  0.01%  0.00% Load Meter
Resource User: Check heaps(ID: 0x1000009)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777225         456       232       1965  0.00%  0.01%  0.00% Check heaps
Resource User: Pool Manager(ID: 0x100000A)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777226           0         1          0  0.00%  0.00%  0.00% Pool Manager
Resource User: Buffer RO RU(ID: 0x100000B)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777227           0         0          0  0.00%  0.00%  0.00% Buffer RO RU
Resource User: Timers(ID: 0x100000C)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777228           0         2          0  0.00%  0.00%  0.00% Timers
.
.
.
Resource Owner: memory
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
Chunk Elements :
Allocated Size(b): 25967632 Count: 46612 Freed Size(b): 21487684 Count: 26053

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
63700E18 0000020052 636FDCD4 63705C6C 001 6412D2C0  Managed Chunk Queue Elements
63705C6C 0000012852 63700E18 63708EA0 001 6412D2C0  *Init*
63708EA0 0000010052 63705C6C 6370B5E4 001 6412D2C0  List Elements
6370B5E4 0000005052 63708EA0 6370C9A0 001 6412D2C0  List Headers
6370C9A0 0000009052 6370B5E4 6370ECFC 001 6412D2C0  Interrupt Stack
6370ECFC 0000000096 6370C9A0 6370ED5C 001 6412D2C0  *Init*
```

```
6370ED5C 0000000084 6370ECFC 6370EDB0 001 6412D2C0  *Init*
6370EDB0 0000000132 6370ED5C 6370EE34 001 6412D2C0  *Init*
6370EE34 0000000092 6370EDB0 6370EE90 001 6412D2C0  *Init*
6370EE90 0000000436 6370EE34 6370F044 001 6412D2C0  *Init*
6370F044 0000000076 6370EE90 6370F090 001 6412D2C0  *Init*
6370F090 0000000132 6370F044 6370F114 001 6412D2C0  *Init*
6370F114 0000000092 6370F090 6370F170 001 6412D2C0  *Init*
.
.
.
Resource User: Scheduler(ID: 0x1000002)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
63799F04 0000012052 63799EB8 6379CE18 001 6412D2C0  Scheduler Stack
643E9A38 0000000076 643D9A04 643E9A84 001 6412D2C0  *Sched*
644C47F0 0000000076 644C4790 644C483C 001 6412D2C0  *Sched*
645FF744 0000000096 645FF6E8 645FF7A4 001 6412D2C0  *Sched*
64904354 0000000112 649040D0 649043C4 001 6412D2C0  *Sched*
  Resource User: Dead(ID: 0x1000003)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
63F9D328 0000000096 63F984D4 63F9D388 001 6412D2C0  AAA MI SG NAME
  Resource User: Interrupt(ID: 0x1000004)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0
```

The following is sample output from the **show resource owner all user all detailed triggers** command:

```
Router# show resource owner all user all detailed triggers

Resource Owner: cpu
 Resource User Type: iosprocess
  Resource User: Init(ID: 0x1000001)
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777217           0         0          0  0.00%  0.00%  0.00% Init
Resource User: Scheduler(ID: 0x1000002)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource User: Dead(ID: 0x1000003)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource User: Interrupt(ID: 0x1000004)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource User: Memory RO RU(ID: 0x1000005)
RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource User: Chunk Manager(ID: 0x1000006)
.
.
.
Resource User: Scheduler(ID: 0x1000002)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0

Processor memory
 Address      Bytes     Prev     Next Ref   Alloc PC  What
63799F04 0000012052 63799EB8 6379CE18 001 6412D2C0  Scheduler Stack
```

```
643E9A38 0000000076 643D9A04 643E9A84 001 6412D2C0  *Sched*
644C47F0 0000000076 644C4790 644C483C 001 6412D2C0  *Sched*
645FF744 0000000096 645FF6E8 645FF7A4 001 6412D2C0  *Sched*
64904354 0000000112 649040D0 649043C4 001 6412D2C0  *Sched*
  Resource User: Dead(ID: 0x1000003)
Chunk Elements :
Allocated Size(b): 0 Count: 0 Freed Size(b): 0 Count: 0
.
.
.
Resource User Type: test_RUT142
Resource User Type: test_RUT143
Resource User Type: test_RUT144
Resource User Type: test_RUT145
Resource User Type: test_RUT146
Resource User Type: test_RUT147
Resource User Type: test_RUT148
Resource User Type: test_RUT149
```

Table 89 describes the significant fields shown in the display.

Table 89 show resource owner Field Descriptions

Field	Description
Runtime(ms)	The runtime of the process in milliseconds.
Invoked	The number of times an RU has been allowed to run.
uSecs	The amount of runtime per invocation in microseconds.
Allocated Size(b)	The number of bytes of memory that are allocated.
Freed Size(b)	The number of bytes of memory that are freed.
Count	The number of elements that are allocated or freed. For example, if two elements of 50 bytes each are allocated, the allocated count is 2 and allocated size is 100.
pc	Displays the details of the memory that is held by a process. Each line of the output displays one or more blocks of memory. The pc is the allocator pc of a particular block of memory.
size	The total size of memory allocated to each block. The sum of the size of all blocks is equivalent to the total memory held by the process.
count	The count is the number of blocks of memory.
Getbufs	The number of buffers allocated by the RU.
Retbufs	The number of buffers freed by the RU.
Holding	The number of buffers the RU is holding currently.

Related Commands

Command	Description
buffer public	Enters buffer owner configuration mode and sets thresholds for buffer usage.
cpu interrupt	Enters CPU owner configuration mode and sets thresholds for interrupt level CPU utilization.
cpu process	Enters CPU owner configuration mode and sets thresholds for processor level CPU utilization.

Command	Description
cpu total	Enters CPU owner configuration mode and sets thresholds for total CPU utilization.
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
memory io	Enters memory owner configuration mode and sets threshold values for I/O memory.
memory processor	Enters memory owner configuration mode and sets threshold values for processor memory.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show resource all	Displays all the resource details.
show resource database	Displays the entire database of all resource entry relationships.
show resource relationship	Displays the relationship between the RUs and the ROs.

show resource relationship

To display the details of relationships between different resource owners, use the **show resource relationship** command in user EXEC or privileged EXEC mode.

show resource relationship [**user** *resource-user-type*]

Syntax Description

user	(Optional) Identifies a resource user (RU).
resource-user-type	(Optional) Type of RU.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show resource relationship** command:

```
Router# show resource relationship

Resource User Type: iosprocess (ID: 0x1)
 -> Resource Owner: cpu (ID: 0x1)
 -> Resource Owner: memory (ID: 0x2)
 -> Resource Owner: Buffer (ID: 0x3)
 -> Resource User: Init (ID: 0x1000001)
 -> Resource User: Scheduler (ID: 0x1000002)
 -> Resource User: Dead (ID: 0x1000003)
 -> Resource User: Interrupt (ID: 0x1000004)
 -> Resource User: Memory RO RU (ID: 0x1000005)
 -> Resource User: Chunk Manager (ID: 0x1000006)
 -> Resource User: Load Meter (ID: 0x1000007)
 -> Resource User: Check heaps (ID: 0x1000009)
 -> Resource User: Pool Manager (ID: 0x100000A)
 -> Resource User: Buffer RO RU (ID: 0x100000B)
 -> Resource User: Timers (ID: 0x100000C)
 -> Resource User: Serial Background (ID: 0x100000D)
 -> Resource User: ALARM_TRIGGER_SCAN (ID: 0x100000E)
 -> Resource User: AAA_SERVER_DEADTIME (ID: 0x100000F)
 -> Resource User: AAA high-capacity counters (ID: 0x1000010)
 -> Resource User: Policy Manager (ID: 0x1000011)
 -> Resource User: Crash writer (ID: 0x1000012)
 -> Resource User: RO Notify Timers (ID: 0x1000013)
 -> Resource User: RMI RM Notify Watched Policy (ID: 0x1000014)
 -> Resource User: EnvMon (ID: 0x1000015)
 -> Resource User: OIR Handler (ID: 0x1000016)
 -> Resource User: IPC Dynamic Cache (ID: 0x1000017)
 -> Resource User: IPC Zone Manager (ID: 0x1000018)
 -> Resource User: IPC Periodic Timer (ID: 0x1000019)
 -> Resource User: IPC Managed Timer (ID: 0x100001A)
```

```
 -> Resource User: IPC Deferred Port Closure (ID: 0x100001B)
 -> Resource User: IPC Seat Manager (ID: 0x100001C)
 -> Resource User: IPC Session Service (ID: 0x100001D)
 -> Resource User: Compute SRP rates (ID: 0x100001E)
 -> Resource User: ARP Input (ID: 0x100001F)
 -> Resource User: DDR Timers (ID: 0x1000020)
 -> Resource User: Dialer event (ID: 0x1000021)
 -> Resource User: Entity MIB API (ID: 0x1000022)
 -> Resource User: SERIAL A'detect (ID: 0x1000023)
 -> Resource User: GraphIt (ID: 0x1000024)
 -> Resource User: HC Counter Timers (ID: 0x1000025)
.
.
.
Resource User Type: test_RUT141 (ID: 0x92)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT142 (ID: 0x93)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT143 (ID: 0x94)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT144 (ID: 0x95)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT145 (ID: 0x96)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT146 (ID: 0x97)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT147 (ID: 0x98)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT148 (ID: 0x99)
 -> Resource Owner: test_RO0 (ID: 0x7)
 Resource User Type: test_RUT149 (ID: 0x9A)
 -> Resource Owner: test_RO0 (ID: 0x7)
```

Related Commands

Command	Description
buffer public	Enters buffer owner configuration mode and sets thresholds for buffer usage.
cpu interrupt	Enters CPU owner configuration mode and sets thresholds for interrupt level CPU utilization.
cpu process	Enters CPU owner configuration mode and sets thresholds for processor level CPU utilization.
cpu total	Enters CPU owner configuration mode and sets thresholds for total CPU utilization.
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
memory io	Enters memory owner configuration mode and sets threshold values for the I/O memory.
memory processor	Enters memory owner configuration mode and sets threshold values for the processor memory.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show resource all	Displays all the resource details.

Command	Description
show resource database	Displays the entire database of all resource entry relationships.
show resource owner	Displays the RO details.

show resource user

To display the policy details or Resource User (RU) template details of a resource user, use the **show resource user** command in user EXEC or privileged EXEC mode.

show resource user {**all** | *resource-user-type*} [**brief** | **detailed**]

Syntax Description

all	Displays the policy details of all the RUs.
resource-user-type	Type of RU. For example, iosprocess.
brief	(Optional) Displays a short description of the policy details.
detailed	(Optional) Displays all details of a policy.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following is sample output from the **show resource user** command:

```
Router# show resource user all

Resource User Type: iosprocess
Resource Grp: Init
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
27197780  8950144 18247636     6552

I/O memory
Allocated   Freed  Holding   Blocks
 7296000     9504  7286496      196

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777224       14408       116     124206 100.40%  8.20%  1.70% Init
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
332      60       272      Init

Resource User: Init
Resource User: Scheduler
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
   77544        0    77544        2

Resource Owner: cpu
```

```
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777218           0         0          0  0.00%  0.00%  0.00% Scheduler
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        Scheduler

Resource User: Dead
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
 1780540      260  1780280      125

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777219           0         0          0  0.00%  0.00%  0.00% Dead
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
9        8        1        Dead

Resource User: Interrupt
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
       0        0        0        0

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777220           0         0          0  0.00%  0.00%  0.00% Interrupt
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
14128    14128    0        Interrupt

Resource User: Memory RO RU
Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
  132560     1480   131080        2

Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777221           0         0          0  0.00%  0.00%  0.00% Memory RO RU
Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
64       64       0        Memory RO RU
.
.
.
Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777401        7124      4250       1676  0.00%  0.03%  0.01% Exec
  Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
38       38       0        Exec

 Resource User: BGP Router
  Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
   43380    26556    16824        8

  Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777404          12     19705          0  0.00%  0.00%  0.00% BGP Router
  Resource Owner: Buffer
```

```
Getbufs  Retbufs  Holding  RU Name
0        0        0        BGP Router

 Resource User: BGP I/O
  Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
    6892     6892        0        0

  Resource Owner: cpu
    RUID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min Res Usr
16777405           0         1          0  0.00%  0.00%  0.00% BGP I/O
  Resource Owner: Buffer
Getbufs  Retbufs  Holding  RU Name
0        0        0        BGP I/O

 Resource User: BGP Scanner
  Resource Owner: memory
Processor memory
Allocated   Freed  Holding   Blocks
    9828     9828        0        0

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

Resource User Type: test_process
Resource User Type: mem_rut
Resource User Type: cpu_rut
```

Table 90 describes the significant fields shown in the display.

Table 90 show resource user Field Descriptions

Field	Description
Allocated	The number of bytes of memory that is allocated.
Freed	The number of bytes of memory that is freed.
Count	The number of elements that are allocated or freed. For example, if two elements of 50 bytes each are allocated, the allocated count is 2 and allocated size is 100.
Runtime(ms)	The runtime of the process in milliseconds.
Invoked	The number of times an RU has been allowed to run.
uSecs	The amount of runtime per invocation in microseconds.
Getbufs	The number of buffers allocated by the RU.
Retbufs	The number of buffers freed by the RU.
Holding	The number of buffers the RU is holding currently.

Related Commands

Command	Description
buffer public	Enters buffer owner configuration mode and sets thresholds for buffer usage.
cpu interrupt	Enters CPU owner configuration mode and sets thresholds for interrupt-level CPU utilization.
cpu process	Enters CPU owner configuration mode and sets thresholds for processor-level CPU utilization.
cpu total	Enters CPU owner configuration mode and sets thresholds for total CPU utilization.
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
memory io	Enters memory owner configuration mode and sets threshold values for I/O memory.
memory processor	Enters memory owner configuration mode and sets threshold values for processor memory.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show resource all	Displays all the resource details.
show resource database	Displays entire database of all resource entry relationships.
show resource owner	Displays the RO details.

show rmon

To display the current RMON agent status on the router, use the **show rmon** command in EXEC mode.

show rmon [**alarms** | **capture** | **events** | **filter** | **history** | **hosts** | **matrix** | **statistics** | **task** | **topn**]

Syntax Description

alarms	(Optional) Displays the RMON alarm table.
capture	(Optional) Displays the RMON buffer capture table. Available on Cisco 2500 series and Cisco AS5200 series only.
events	(Optional) Displays the RMON event table.
filter	(Optional) Displays the RMON filter table. Available on Cisco 2500 series and Cisco AS5200 series only.
history	(Optional) Displays the RMON history table. Available on Cisco 2500 series and Cisco AS5200 series only.
hosts	(Optional) Displays the RMON hosts table. Available on Cisco 2500 series and Cisco AS5200 series only.
matrix	(Optional) Displays the RMON matrix table. Available on Cisco 2500 series and Cisco AS5200 series only.
statistics	(Optional) Displays the RMON statistics table. Available on Cisco 2500 series and Cisco AS5200 series only.
task	(Optional) Displays general RMON statistics. This is the default.
topn	(Optional) Displays the RMON top-n hosts table. Available on Cisco 2500 series and Cisco AS5200 series only.

Command Default

If no option is specified, the **task** option is displayed.

Command Modes

EXEC

Command History

Release	Modification
11.1	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Refer to the specific **show rmon** command for an example and description of the fields.

For additional information, refer to the RMON MIB described in RFC 1757.

Examples

The following is sample output from the **show rmon** command. All counters are from the time the router was initialized.

```
Router# show rmon

145678 packets input (34562 promiscuous), 0 drops
145678 packets processed, 0 on queue, queue utilization 15/64
```

Table 91 describes the significant fields shown in the ouput.

Table 91 show rmon Field Descriptions

Field	Description
x packets input	Number of packets received on RMON-enabled interfaces.
x promiscuous	Number of input packets that were seen by the router only because RMON placed the interface in promiscuous mode.
x drops	Number of input packets that could not be processed because the RMON queue overflowed.
x packets processed	Number of input packets actually processed by the RMON task.
x on queue	Number of input packets that are sitting on the RMON queue, waiting to be processed.
queue utilization *x*/*y*	*y* is the maximum size of the RMON queue; *x* is the largest number of packets that were ever on the queue at a particular time.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
rmon queuesize	Changes the size of the queue that holds packets for analysis by the RMON process.
show rmon alarms	Displays the contents of the router's RMON alarm table.
show rmon capture	Displays the contents of the router's RMON capture table.
show rmon events	Displays the contents of the router's RMON event table.
show rmon filter	Displays the contents of the router's RMON filter table.
show rmon history	Displays the contents of the router's RMON history table.
show rmon hosts	Displays the contents of the router's RMON hosts table.
show rmon matrix	Displays the contents of the router's RMON matrix table.
show rmon statistics	Displays the contents of the router's RMON statistics table.
show rmon topn	Displays the contents of the router's RMON p-N host table.

show rmon alarms

To display the contents of the RMON alarm table of the router, use the **show rmon alarms** command in EXEC mode.

show rmon alarms

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON alarms to display alarm information with the **show rmon alarms** command.

Examples

The following is sample output from the **show rmon alarms** command:

```
Router# show rmon alarms

Alarm 2 is active, owned by manager1
 Monitors ifEntry.1.1 every 30 seconds
 Taking delta samples, last value was 0
 Rising threshold is 15, assigned to event 12
 Falling threshold is 0, assigned to event 0
 On startup enable rising or falling alarm
```

Table 92 describes the significant fields shown in the display.

Table 92 show rmon alarms Field Descriptions

Field	Description
Alarm 2 is active, owned by manager1	Unique index into the alarmTable, showing the alarm status is active, and the owner of this row, as defined in the alarmTable of RMON.
Monitors ifEntry.1.1	Object identifier of the particular variable to be sampled. Equivalent to alarmVariable in RMON.

Table 92 show rmon alarms Field Descriptions (continued)

Field	Description
every 30 seconds	Interval in seconds over which the data is sampled and compared with the rising and falling thresholds. Equivalent to alarmInterval in RMON.
Taking delta samples	Method of sampling the selected variable and calculating the value to be compared against the thresholds. Equivalent to alarmSampleType in RMON.
last value was	Value of the statistic during the last sampling period. Equivalent to alarmValue in RMON.
Rising threshold is	Threshold for the sampled statistic. Equivalent to alarmRisingThreshold in RMON.
assigned to event	Index of the eventEntry that is used when a rising threshold is crossed. Equivalent to alarmRisingEventIndex in RMON.
Falling threshold is	Threshold for the sampled statistic. Equivalent to alarmFallingThreshold in RMON.
assigned to event	Index of the eventEntry that is used when a falling threshold is crossed. Equivalent to alarmFallingEventIndex in RMON.
On startup enable rising or falling alarm	Alarm that may be sent when this entry is first set to valid. Equivalent to alarmStartupAlarm in RMON.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
show rmon	Displays the current RMON agent status on the router.

show rmon capture

To display the contents of the router's RMON capture table, use the **show rmon capture** command in EXEC mode.

show rmon capture

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON alarms and events to display alarm information with the **show rmon capture** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

Examples

The following is sample output from the **show rmon capture** command:

```
Router# show rmon capture

Buffer 4096 is active, owned by manager1
 Captured data is from channel 4096
 Slice size is 128, download size is 128
 Download offset is 0
 Full Status is spaceAvailable, full action is lockWhenFull
 Granted 65536 octets out of 65536 requested
 Buffer has been on since 00:01:16, and has captured 1 packets
  Current capture buffer entries:
   Packet 1 was captured 416 ms since buffer was turned on
   Its length is 326 octets and has a status type of 0
   Packet ID is 634, and contains the following data:
00 00 0c 03 12 ce 00 00 0c 08 9d 4e 08 00 45 00
01 34 01 42 00 00 1d 11 e3 01 ab 45 30 15 ac 15
31 06 05 98 00 a1 01 20 9f a8 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
```

Table 93 describes the significant fields shown in the ouput.

Table 93 show rmon capture Field Descriptions

Field	Description
Buffer 4096 is active	Equates to bufferControlIndex in the bufferControlTable of RMON. Uniquely identifies a valid (active) row in this table.
owned by manager1	Denotes the owner of this row. Equates to bufferControlOwner in the bufferControlTable of RMON.
Captured data is from channel	Equates to the bufferControlChannelIndex and identifies which RMON channel is the source of these packets.
Slice size is	Identifies the maximum number of octets of each packet that will be saved in this capture buffer. Equates to bufferControlCaptureSliceSize of RMON.
download size is	Identifies the maximum number of octets of each packet in this capture buffer that will be returned in an SNMP retrieval of that packet. Equates to bufferControlDownloadSliceSize in RMON.
Download offset is	Offset of the first octet of each packet in this capture buffer that will be returned in an SNMP retrieval of that packet. Equates to bufferControlDownloadOffset in RMON.
Full Status is spaceAvailable	Shows whether the buffer is full or has room to accept new packets. Equates to bufferControlFullStatus in RMON.
full action is lockWhenFull	Controls the action of the buffer when it reaches full status. Equates to bufferControlFullAction in RMON.
Granted 65536 octets	Actual maximum number of octets that can be saved in this capture buffer. Equates to bufferControlMaxOctetsGranted in RMON.
out of 65536 requested	Requested maximum number of octets to be saved in this capture buffer. Equates to bufferControlMaxOctetsRequested in RMON.
Buffer has been on since	Indicates how long the buffer has been available.
and has captured 1 packets	Number of packets captured since buffer was turned on. Equates to bufferControlCapturedPackets in RMON.
Current capture buffer entries:	Lists each packet captured.
Packet 1 was captured 416 ms since buffer was turned on Its length is 326 octets and has a status type of 0	Zero indicates the error status of this packet. Equates to captureBufferPacketStatus in RMON, where its value options are documented.
Packet ID is	Index that describes the order of packets received on a particular interface. Equates to captureBufferPacketID in RMON.
and contains the following data:	Data inside the packet, starting at the beginning of the packet.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon events

To display the contents of the router's RMON event table, use the **show rmon events** command in EXEC mode.

show rmon events

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON events to display alarm information with the **show rmon events** command.

Examples

The following is sample output from the **show rmon events** command:

```
Router# show rmon events

Event 12 is active, owned by manager1
 Description is interface-errors
 Event firing causes log and trap to community rmonTrap, last fired 00:00:00
```

Table 94 describes the significant fields shown in the display.

Table 94 show rmon events Field Descriptions

Field	Description
Event 12 is active, owned by manager1	Unique index into the eventTable, showing the event status is active, and the owner of this row, as defined in the eventTable of RMON.
Description is interface-errors	Type of event, in this case an interface error.
Event firing causes log and trap	Type of notification that the router will make about this event. Equivalent to eventType in RMON.

Table 94 show rmon events Field Descriptions (continued)

Field	Description
community rmonTrap	If an SNMP trap is to be sent, it will be sent to the SNMP community specified by this octet string. Equivalent to eventCommunity in RMON.
last fired	Last time the event was generated.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon filter

To display the contents of a router's Remote Monitoring (RMON) filter table, use the **show rmon filter** command in privileged EXEC mode.

show rmon filter

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, see the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface and configured RMON alarms and events to display alarm information with the **show rmon filter** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

Examples

The following is sample output from the **show rmon filter** command:

```
Router# show rmon filter

Filter 4096 is active, and owned by manager1
 Data offset is 12, with
 Data of  08 00 00 00 00 00 00 00 00 00 00 00 00 00 ab 45 30 15 ac 15 31 06
 Data Mask is ff ff 00 00 00 00 00 00 00 00 00 00 00 00 ff ff ff ff ff ff ff ff
 Data Not Mask is 0
 Pkt status is 0, status mask is 0, not mask is 0
 Associated channel 4096 is active, and owned by manager1
 Type of channel is acceptFailed, data control is off
 Generate event index 0
 Event status is eventFired, # of matches is 1482
 Turn on event index is 0, turn off event index is 0
 Description:
```

Table 95 describes the significant fields shown in the display.

Table 95 show rmon filter Field Descriptions

Field	Description
Filter *x* is active, and owned by *y*	Unique index of the filter, its current state, and the owner, as defined in the filterTable of RMON.
Data offset is	Offset from the beginning of each packet where a match of packet data will be attempted. Equivalent to filterPktDataOffset in RMON.
Data of	Data that is to be matched with the input packet. Equivalent to filterPktData in RMON.
Data Mask is	Mask that is applied to the match process. Equivalent to filterPktDataMask in RMON.
Data Not Mask is	Inversion mask that is applied to the match process. Equivalent to filterPktDataNotMask in RMON.
Pkt status is	Status that is to be matched with the input packet. Equivalent to filterPktStatus in RMON.
status mask is	Mask that is applied to the status match process. Equivalent to filterPktStatusMask in RMON.
not mask is	Inversion mask that is applied to the status match process. Equivalent to filterPktStatusNotMask in RMON.
Associated channel *x* is active, and owned by *y*	Unique index of the channel, its current state, and the owner, as defined in the channelTable of RMON.
Type of channel is {acceptMatched | acceptFailed}	This object controls the action of the filters associated with this channel. Equivalent to channelAcceptType of RMON.
data control is {off | on }	This object controls the flow of data through this channel. Equivalent to channelDataControl in RMON.
Generate event index 0	Value of this object identifies the event that is to be generated when the associated channelDataControl is on and a packet is matched. Equivalent to channelEventIndex in RMON.
Event status is eventFired	When the channel is configured to generate events and when packets are matched, this message indicates the means of controlling the flow of those events. Equivalent to channelEventStatus in RMON.
# of matches is	Number of times this channel has matched a packet. Equivalent to channelMatches in RMON.
Turn on event index is	Value of this object identifies the event that is configured to turn the associated channelDataControl from off to on when the event is generated. Equivalent to channelTurnOnEventIndex in RMON.

Table 95 show rmon filter Field Descriptions (continued)

Field	Description
Turn off event index is	Value of this object identifies the event that is configured to turn the associated channelDataControl from on to off when the event is generated. Equivalent to channelTurnOffEventIndex in RMON.
Description:	Comment describing this channel.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon hc-alarms

To display the contents of the RMON high-capacity (HC) alarm table of the router, use the **show rmon hc-alarms** command in user EXEC or privileged EXEC mode.

show rmon hc-alarms

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC (>)
Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXI	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

For additional information on RMON MIB, refer to the RMON MIB for HC Alarms described in RFC 3434.

You must first enable RMON on the interface and configure RMON HC alarms before you can display alarm information with the **show rmon hc-alarms** command.

Examples

The following is sample output from the **show rmon hc-alarms** command:

```
Router# show rmon hc-alarms

Monitors ifInOctets.1 every 20 second(s)
 Taking absolute samples, last value was 0
 Rising threshold Low is 4096, Rising threshold Hi is 0,
               assigned to event 0
 Falling threshold Low is 1280, Falling threshold Hi is 0,
               assigned to event 0
 On startup enable rising or falling alarm
```

Table 92 describes the fields shown in the display.

Table 96 show rmon hc-alarms Field Descriptions

Field	Description
Monitors ifInOctets.1	Object identifier of the particular variable to be sampled. Equivalent to alarmVariable in RMON.
every 20 seconds	Interval in seconds over which the data is sampled and compared with the rising and falling thresholds. Equivalent to alarmInterval in RMON.

Table 96 show rmon hc-alarms Field Descriptions (continued)

Field	Description
Taking absolute samples	Method of sampling the selected variable and calculating the value to be compared against the thresholds. Equivalent to alarmSampleType in RMON.
last value was	Value of the statistic during the last sampling period. Equivalent to alarmValue in RMON.
Rising threshold Low is Rising threshold High is	Threshold for the sampled statistic. Equivalent to alarmRisingThreshold in RMON.
assigned to event	Index of the eventEntry that is used when a rising threshold is crossed. Equivalent to alarmRisingEventIndex in RMON.
Falling threshold Low is Falling threshold High is	Threshold for the sampled statistic. Equivalent to alarmFallingThreshold in RMON.
assigned to event	Index of the eventEntry that is used when a falling threshold is crossed. Equivalent to alarmFallingEventIndex in RMON.
On startup enable rising or falling alarm	Alarm that may be sent when this entry is first set to valid. Equivalent to alarmStartupAlarm in RMON.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon hc-alarms	Sets a high-capacity alarm on any MIB object.

show rmon history

To display the contents of the router's RMON history table, use the **show rmon history** command in EXEC mode.

show rmon history

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON alarms and events to display alarm information with the **show rmon history** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

Examples

The following is sample output from the **show rmon history** command:

```
Router# show rmon history

Entry 1 is active, and owned by manager1
 Monitors ifEntry.1.1 every 30 seconds
 Requested # of time intervals, ie buckets, is 5
 Granted # of time intervals, ie buckets, is 5
  Sample # 14 began measuring at 00:11:00
   Received 38346 octets, 216 packets,
   0 broadcast and 80 multicast packets,
   0 undersized and 0 oversized packets,
   0 fragments and 0 jabbers,
   0 CRC alignment errors and 0 collisions.
   # of dropped packet events is 0
   Network utilization is estimated at 10
```

Table 97 describes the significant fields shown in the display.

Table 97 show rmon history Field Descriptions

Field	Description
Entry 1 is active, and owned by manager1	Unique index of the history entry, its current state, and the owner as defined in the historyControlTable of RMON.
Monitors ifEntry.1.1	This object identifies the source of the data for which historical data was collected and placed in a media-specific table. Equivalent to historyControlDataSource in RMON.
every 30 seconds	Interval in seconds over which the data is sampled for each bucket in the part of the media-specific table associated with this historyControlEntry. Equivalent to historyControlInterval in RMON.
Requested # of time intervals, ie buckets, is	Requested number of discrete time intervals over which data is to be saved in the part of the media-specific table associated with this historyControlEntry. Equivalent to historyControlBucketsRequested in RMON.
Granted # of time intervals, ie buckets, is	Actual number of discrete time intervals over which data is to be saved in the part of the media-specific table associated with this historyControlEntry. Equivalent to historyControlBucketsGranted in RMON.
Sample # 14 began measuring at	Time at the start of the interval over which this sample was measured.
Received 38346 octets	Total number of octets of data (including those in bad packets) received on the network (excluding framing bits but including FCS octets). Equivalent to etherHistoryOctets in RMON.
x packets	Number of packets (including bad packets) received during this sampling interval. Equivalent to etherHistoryPkts in RMON.
x broadcast	Number of good packets received during this sampling interval that were directed to the broadcast address. Equivalent to etherHistoryBroadcastPkts in RMON.
x multicast packets	Number of good packets received during this sampling interval that were directed to a multicast address. Equivalent to etherHistoryMulticastPkts in RMON.
x undersized	Number of packets received during this sampling interval that were fewer than 64 octets long (excluding framing bits but including FCS octets) and were otherwise well formed. Equivalent to etherHistoryUndersizedPkts in RMON.
x oversized packets	Number of packets received during this sampling interval that were longer than 1518 octets (excluding framing bits but including FCS octets) but were otherwise well formed. Equivalent to etherHistoryOversizePkts in RMON.

Table 97 show rmon history Field Descriptions (continued)

Field	Description
x fragments	Total number of packets received during this sampling interval that were fewer than 64 octets in length (excluding framing bits but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Equivalent to etherHistoryFragments in RMON.
x jabbers	Number of packets received during this sampling interval that were longer than 1518 octets (excluding framing bits but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Note that this definition of jabber is different than the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). Equivalent to etherHistoryJabbers in RMON.
x CRC alignment errors	Number of packets received during this sampling interval that had a length (excluding framing bits but including FCS octets) from 64 to 1518 octets, inclusive, but had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Equivalent to etherHistoryCRCAlignErrors in RMON.
x collisions	Best estimate of the total number of collisions on this Ethernet segment during this sampling interval. Equivalent to etherHistoryCollisions in RMON.
# of dropped packet events is	Total number of events in which packets were dropped by the operation because of resources during this sampling interval. Note that this number is not necessarily the number of packets dropped, it is just the number of times this condition has been detected. Equivalent to etherHistoryDropEvents in RMON.
Network utilization is estimated at	Best estimate of the mean physical-layer network usage on this interface during this sampling interval, in hundredths of a percent. Equivalent to etherHistoryUtilization in RMON.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon hosts

To display the contents of the router's RMON hosts table, use the **show rmon hosts** command in EXEC mode.

show rmon hosts

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

You must have first enabled RMON on the interface, and configured RMON alarms and events to display alarm information with the **show rmon hosts** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

For additional information, refer to the RMON MIB described in RFC 1757.

Examples

The following is sample output from the **show rmon hosts** command:

```
Router# show rmon hosts

Host Control Entry 1 is active, and owned by manager1
 Monitors host ifEntry.1.1
 Table size is 51, last time an entry was deleted was 00:00:00
  Creation Order number is 1
   Physical address is 0000.0c02.5808
   Packets: rcvd 6963, transmitted 7041
   Octets: rcvd 784062, transmitted 858530
   # of packets transmitted: broadcast 28, multicast 48
   # of bad packets transmitted is 0
```

Table 98 describes the significant fields shown in the display.

Table 98 show rmon hosts Field Descriptions

Field	Description
Host Control Entry 1 is active, and owned by manager1	Unique index of the host entry, its current state, and the owner as defined in the hostControlTable of RMON.
Monitors host ifEntry.1.1	This object identifies the source of the data for this instance of the host function. Equivalent to hostControlDataSource in RMON.
Table size is	Number of hostEntries in the hostTable and the hostTimeTable associated with this hostControlEntry. Equivalent to hostControlTableSize in RMON.
last time an entry was deleted was	Time when the last entry was deleted from the hostTable.
Creation Order number is	Index that defines the relative ordering of the creation time of hosts captured for a particular hostControlEntry. Equivalent to hostCreationOrder in RMON.
Physical address is	Physical address of this host. Equivalent to hostAddress in RMON.
Packets: rcvd	Number of good packets transmitted to this address. Equivalent to hostInPkts in RMON.
transmitted	Number of packets, including bad packets transmitted by this address. Equivalent to hostOutPkts in RMON.
Octets: rcvd	Number of octets transmitted to this address since it was added to the hostTable (excluding framing bits but including FCS octets), except for those octets in bad packets. Equivalent to hostInOctets in RMON.
transmitted	Number of octets transmitted by this address since it was added to the hostTable (excluding framing bits but including FCS octets), including those octets in bad packets. Equivalent to hostOutOctets in RMON.
# of packets transmitted:	Number of good packets transmitted by this address that were broadcast or multicast.
# of bad packets transmitted is	Number of bad packets transmitted by this address.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon matrix

To display the contents of the router's RMON matrix table, use the **show rmon matrix** command in EXEC mode.

show rmon matrix

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

You must have first enabled RMON on the interface, and configured RMON alarms and events to display alarm information with the **show rmon matrix** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

For additional information, refer to the RMON MIB described in RFC 1757.

Examples

The following is sample output from the **show rmon matrix** command:

```
Router# show rmon matrix

Matrix 1 is active, and owned by manager1
 Monitors ifEntry.1.1
 Table size is 451, last time an entry was deleted was at 00:00:00
```

Table 99 describes the significant fields shown in the display.

Table 99 show rmon matrix Field Descriptions

Field	Description
Matrix 1 is active, and owned by manager1	Unique index of the matrix entry, its current state, and the owner as defined in the matrixControlTable of RMON.
Monitors ifEntry.1.1	This object identifies the source of the data for this instance of the matrix function. Equivalent to matrixControlDataSource in RMON.
Table size is 451, last time an entry was deleted was at	Size of the matrix table and the time that the last entry was deleted.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon statistics

To display the contents of the router's RMON statistics table, use the **show rmon statistics** command in EXEC mode.

show rmon statistics

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON alarms and events to display alarm information with the **show rmon statistics** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

Examples

The following is sample output from the **show rmon statistics** command:

```
Router# show rmon statistics

Interface 1 is active, and owned by config
 Monitors ifEntry.1.1 which has
 Received 60739740 octets, 201157 packets,
 1721 broadcast and 9185 multicast packets,
 0 undersized and 0 oversized packets,
 0 fragments and 0 jabbers,
 0 CRC alignment errors and 32 collisions.
 # of dropped packet events (due to lack of resources): 511
 # of packets received of length (in octets):
  64: 92955, 65-127: 14204, 128-255: 1116,
  256-511: 4479, 512-1023: 85856, 1024-1518:2547
```

Table 100 describes the significant fields shown in the display.

Table 100 show rmon statistics Field Descriptions

Field	Description
Interface 1 is active, and owned by config	Unique index of the statistics entry, its current state, and the owner as defined in the etherStatsTable of RMON.
Monitors ifEntry.1.1	This object identifies the source of the data that this etherStats entry is configured to analyze. Equivalent to etherStatsDataSource in RMON.
Received 60739740 octets	Total number of octets of data (including those in bad packets) received on the network (excluding framing bits but including FCS octets). Equivalent to etherStatsOctets in RMON.
x packets	Number of packets (including bad packets) received. Equivalent to etherStatsPkts in RMON.
x broadcast	Number of good packets received that were directed to the broadcast address. Equivalent to etherStatsBroadcastPkts in RMON.
x multicast packets	Number of good packets received that were directed to a multicast address. Equivalent to etherStatsMulticastPkts in RMON.
x undersized	Number of packets received that were fewer than 64 octets long (excluding framing bits but including FCS octets) and were otherwise well formed. Equivalent to etherStatsUndersizedPkts in RMON.
x oversized packets	Number of packets received that were longer than 1518 octets (excluding framing bits but including FCS octets) but were otherwise well formed. Equivalent to etherStatsOversizePkts in RMON.
x fragments	Total number of packets received that were fewer than 64 octets in length (excluding framing bits but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Equivalent to etherStatsFragments in RMON.
x jabbers	Number of packets received that were longer than 1518 octets (excluding framing bits but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Note that this definition of jabber is different than the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). Equivalent to etherStatsJabbers in RMON.

Table 100 show rmon statistics Field Descriptions (continued)

Field	Description
x CRC alignment errors	Number of packets received that had a length (excluding framing bits but including FCS octets) from 64 to 1518 octets, inclusive, but had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). Equivalent to etherStatsCRCAlignErrors in RMON.
x collisions	Best estimate of the total number of collisions on this Ethernet segment. Equivalent to etherHistoryCollisions in RMON.
# of dropped packet events (due to lack of resources):	Total number of events in which packets were dropped by the operation because of a lack of resources. Note that this number is not necessarily the number of packets dropped, it is just the number of times this condition has been detected. Equivalent to etherStatsDropEvents in RMON.
# of packets received of length (in octets):	Separates the received packets (good and bad) by packet size in the given ranges (64, 65 to 127,128 to 255, 256 to 511, 512 to 1023, 1024 to 1516).

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show rmon topn

To display the contents of the router's RMON Top-N host table, use the **show rmon topn** command in EXEC mode.

show rmon topn

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

For additional information, refer to the RMON MIB described in RFC 1757.

You must have first enabled RMON on the interface, and configured RMON events to display alarm information with the **show rmon events** command.

This command is available on the Cisco 2500 series and Cisco AS5200 series only.

Examples

The following is sample output from the **show rmon topn** command:

```
Router# show rmon topn

Host Entry 1 of report 1 is active, owned by manager1
 The rate of change is based on hostTopNInPkts
 This report was last started at 00:00:00
 Time remaining in this report is 0 out of 0
 Hosts physical address is 00ad.beef.002b
 Requested # of hosts: 10, # of hosts granted: 10
Report # 1 of Top N hosts entry 1 is recording
Host 0000.0c02.5808 at a rate of 12
```

Table 101 describes the significant fields shown in the display.

Table 101 show rmon topn Field Descriptions

Field	Description
Host Entry 1 of report 1 is active, owned by manager1	Unique index of the hostTopN entry, its current state, and the owner as defined in the hostTopNControlTable of RMON.
The rate of change is based on hostTopNInPkts	Variable for each host that the hostTopNRate variable is based on.
This report was last started at	Time the report was started.
Time remaining in this report is	Number of seconds left in the report currently being collected. Equivalent to hostTopNTimeRemaining in RMON.
out of	Number of seconds that this report has collected during the last sampling interval, or if this report is currently being collected, the number of seconds that this report is being collected during this sampling interval. Equivalent to hostTopNDuration in RMON.
Hosts physical address is	Host address.
Requested # of hosts:	Maximum number of hosts requested for the Top-N table. Equivalent to hostTopNRequestedSize in RMON.
# of hosts granted:	Maximum number of hosts granted for the Top-N table.Eqivalent to hostTopNGrantedSiz in RMON.
Report # 1 of Top N hosts entry 1 is recording	Report number and entry.
Host 0000.0c02.5808 at a rate of	Physical address of the host, and the amount of change in the selected variable during this sampling interval. Equivalent to hostTopNAddress and hostTopNRate in RMON.

Related Commands

Command	Description
rmon	Enables RMON on an Ethernet interface.
rmon alarm	Sets an alarm on any MIB object.
rmon event	Adds or removes an event in the RMON event table that is associated with an RMON event number.
show rmon	Displays the current RMON agent status on the router.

show shell environment

To display shell environment information about user-defined functions, built-in functions, and user-created variables, use the **show shell environment** command in privileged EXEC mode.

show shell environment

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.

Usage Guidelines

Use this command to display the shell variables used on the router.

Examples

The following example displays both the shell environment variables and shell functions. Field names are self-explanatory

```
Router# show shell environment

# Environment Variables:
# User Environment Variables:
?=0
VAR1=value1
VAR2=value2
prc_change_mode=PRC_IMMEDIATE
prc_change_type=PRC_CONFIG_CHANGE
prc_error_code=PRC_SUCCESS
prc_failure_type=PRC_INVALID
prc_ha_sync=PRC_HA_SYNC
# Global Environment Variables:
# Builtin Environment Variables:
PATH=CLI%Userfunctions%Builtins%SYSTEM
# Environment Functions:

# User Environment Functions:

Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}

Function namespace: DEFAULT
```

```
function enable_logging()
{
 configure terminal
 archive
 logging enable
 record rc
 end
}

# Global Environment Functions:

# Builtin Environment Functions:

Function namespace: DEFAULT
((              evaluate a numeric test expression

Function namespace: DEFAULT
[[              evaluate a logical test expression

Function namespace: DEFAULT
cat             output data from a pipe or file to the terminal

Function namespace: DEFAULT
cut             edit piped output

Function namespace: DEFAULT
echo            echo arguments to the terminal

Function namespace: DEFAULT
false           return false in while or if expressions, and set the result

Function namespace: DEFAULT
fetch           return values from the configuration database

Function namespace: DEFAULT
grep            search for regular expressions in piped output or files

Function namespace: DEFAULT
head            print the first lines in the input

Function namespace: DEFAULT
interface       print interfaces that match the argument

Function namespace: DEFAULT
let             evaluate a numeric expression, and set the result

Function namespace: DEFAULT
man             print information for builtins

Function namespace: DEFAULT
more            page piped output to the terminal

Function namespace: DEFAULT
nl              number the lines in the input

Function namespace: DEFAULT
null            ignore the input

Function namespace: DEFAULT
printf          output formatted data to the terminal

Function namespace: DEFAULT
read            read input into variables
```

```
Function namespace: DEFAULT
set_oper        set operational values

Function namespace: DEFAULT
sleep           pause execution of the terminal

Function namespace: DEFAULT
sort            sort the input

Function namespace: DEFAULT
tail            print the tail of the input

Function namespace: DEFAULT
true            return true in while or if expressions, and set the result

Function namespace: DEFAULT
uname           print system information

Function namespace: DEFAULT
wc              count lines, words, and chars
```

Related Commands

Command	Description
shell environment load	Downloads Cisco IOS.sh environment from a specified file to the current TTY.

show shell functions

To display information about Cisco IOS Shell (IOS.sh) user-defined and built-in functions, use the **show shell functions** command in privileged EXEC mode.

show shell functions [*functionname* | **brief]**

Syntax Description

functionname	(Optional) Specifies a user-defined shell function.
brief	(Optional) Lists the names of the user-defined and built-in functions.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.

Usage Guidelines

Use this command to display the shell information for the router.

Examples

The following example shows how to use the **show shell functions** privileged EXEC command to display the details of the user-defined and built-in functions. Field names are self-explanatory.

```
Router# show shell functions
#User defined functions:

Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}

Function namespace: DEFAULT
function enable_logging()
{
 configure terminal
 archive
 logging enable
 record rc
 end
}
#Global User functions:

#Built-in functions:

Function namespace: DEFAULT
((              evaluate a numeric test expression
```

```
Function namespace: DEFAULT
[[              evaluate a logical test expression

Function namespace: DEFAULT
cat             output data from a pipe or file to the terminal

Function namespace: DEFAULT
cut             edit piped output

Function namespace: DEFAULT
echo            echo arguments to the terminal

Function namespace: DEFAULT
false           return false in while or if expressions, and set the result

Function namespace: DEFAULT
fetch           return values from the configuration database

Function namespace: DEFAULT
grep            search for regular expressions in piped output or files

Function namespace: DEFAULT
head            print the first lines in the input

Function namespace: DEFAULT
interface       print interfaces that match the argument

Function namespace: DEFAULT
let             evaluate a numeric expression, and set the result

Function namespace: DEFAULT
man             print information for builtins

Function namespace: DEFAULT
more            page piped output to the terminal

Function namespace: DEFAULT
nl              number the lines in the input

Function namespace: DEFAULT
null            ignore the input

Function namespace: DEFAULT
printf          output formatted data to the terminal

Function namespace: DEFAULT
read            read input into variables

Function namespace: DEFAULT
set_oper        set operational values

Function namespace: DEFAULT
sleep           pause execution of the terminal

Function namespace: DEFAULT
sort            sort the input

Function namespace: DEFAULT
tail            print the tail of the input

Function namespace: DEFAULT
true            return true in while or if expressions, and set the result
```

```
Function namespace: DEFAULT
uname           print system information

Function namespace: DEFAULT
wc              count lines, words, and chars
```

The following example shows how to use the **show shell functions brief** privileged EXEC command to display a list of the names of user-defined and built-in functions:

```
Router# show shell functions brief

#User defined functions:
enable_archive
enable_logging

#Global User functions:

#Built-in functions:

((
[[
cat
cut
echo
false
fetch
grep
head
interface
let
man
more
nl
null
printf
read
set_oper
sleep
sort
tail
true
uname
wc
```

Table 102 Built-in Cisco IOS.sh Functions

Built-in Function	Description
cat	Output data from a pipe or file to the terminal.
cut	Edit piped output.
echo	Echo arguments to the terminal.
false	Return false in while or if expressions, and set the result.
fetch	Return values from the configuration database.
for	Cisco IOS.sh for loops.
grep	Search for regular expressions in piped output or files.
head	Print the first lines in the input.
interface	Print interfaces that match the argument.

Table 102 Built-in Cisco IOS.sh Functions (continued)

Built-in Function	Description
let	Evaluate a numeric expression, and set the result.
man	Print information for built-ins.
more	Page piped output to the terminal.
nl	Number the lines in the input.
null	Ignore the input.
printf	Output formatted data to the terminal.
read	Read input into variables.
set_oper	Set operational values.
sleep	Pause execution of the terminal.
sort	Sort the input.
tail	Print the tail of the input.
true	Return true in while or if expressions, and set the result.
uname	Print system information.
wc	Count lines, words, and chars.

The following example shows how to use the **show shell functions enable_archive** privileged EXEC command to view the user-defined **enable_archive** function:

```
Router# show shell functions enable_archive
#User defined functions:

Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}
```