Troubleshooting the Cisco IOS XR Software

This chapter describes the tools and procedures used to identify the source of hardware and software problems. This information can be used to resolve an issue or see more specific troubleshooting guides for help on a specific topic. This chapter also provides instructions on gathering data for further analysis by Cisco customer support representatives.

Contents

This chapter contains the following sections:

Additional Sources for Information

For additional information on troubleshooting, see the following sources:

- If the Cisco IOS XR software does not start and display the EXEC mode prompt, see Appendix A, "Router Recovery and Management with ROM Monitor."
- The Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.

  http://www.cisco.com/public/support/tac/home.shtml
- The "Related Documents" section on page xv.

Basic Troubleshooting Commands

The following sections describe some basic techniques used to determine connectivity to another device and display information on the configuration and operation of a router.

Using show Commands to Display System Status and Configuration

Use **show** commands to check the status of various Cisco IOS XR software subsystems and services. All **show** commands (except **show configuration**) are entered in EXEC mode. Table 7-1 lists some of the common **show** commands.

To display a complete list of the available show commands, enter the **show ?** command to access the on-screen help system.

Table 7-1 Common show Commands in Cisco IOS XR Software

Command	Description
show variables boot	Displays the boot variables. **Note** This command must be entered in admin EXEC mode.
show configuration	Displays the uncommitted configuration changes made during a configuration session. This command can be entered in any configuration mode.
show context (and show exception)	Displays context information about all recent reloads.
show controller	Displays hardware controller information.
show debug	Displays debug flags enabled from the current terminal.
show environment [**all** | **fans** | **leds** | **power-supply** | **table** | **temperatures** | **voltages** | **l**]	Displays hardware information for the physical components and systems, including fans, LEDs, power supply voltage and current information, and temperatures.
show exception	Displays all exception dump configurations.
show install	Displays installed and active software packages.
show interfaces	Displays interface status and configuration.
show logging	Displays the contents of logging buffers.
show memory	Displays memory statistics.
show platform	Displays information about node status on the router.
show process blocked	Displays blocked processes.
show redundancy	Display the status of the primary (active) route processor (RP) and the standby (redundant) RP.
show running-config [*command*]	Displays the current running configuration. This command can be entered in either configuration or EXEC mode.

Table 7-1 Common show Commands in Cisco IOS XR Software (continued)

Command	Description
show tech-support	Collects a large amount of system information for troubleshooting. The output should be provided to technical support representatives when a problem is reported. Because of the impact the command can have on a running system, it is reserved for users assigned to the cisco-support task ID.
show tracebacks	Allows you to get a dump of all unsolicited debug messages that contained traceback information (that is, a list of program counters back from the line of code generating the message) along with the dynamic link library (DLL) information necessary to decode the traceback.
show user [**group** | **tasks** | **all**]	Displays the username for the current logged-in user. Use this command to also display the groups and associated task IDs assigned to the account.
show version	Displays basic system information.

Using the ping Command

Use the **ping** command to diagnose network connectivity. Enter a hostname or an IP address as an argument to this command.

The **ping** command sends an echo request packet to an address, then awaits a reply. Ping output can help you evaluate path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

Each exclamation point (!) indicates receipt of a reply. A period (.) indicates the network server timed out while waiting for a reply. Other characters may appear in the ping output display, depending on the protocol type.

Examples

In the following example, a successful ping attempt is shown:

```
RP/0/RP0/CPU0:router# ping 10.233.233.233

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.233.233.233, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/7 ms
```

In the following example, an unsuccessful ping attempt is shown:

```
RP/0/RP0/CPU0:router# ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Using the traceroute Command

Use the **traceroute** command to discover the routes that packets take when traveling to their destination. Enter a hostname or an IP address as an argument to this command.

This command works by taking advantage of the error messages generated by routers when a datagram exceeds its time-to-live (TTL) value.

The **traceroute** command starts by sending probe datagrams with a TTL value of 1, causing the first router to discard the probe datagram and send back an error message. The **traceroute** command sends several probes at each TTL level and displays the round-trip time for each.

The **traceroute** command sends one probe at a time. Each outgoing packet may result in one or two error messages. A *time exceeded* error message indicates that an intermediate router has seen and discarded the probe. A *destination unreachable* error message indicates that the destination node has received the probe and discarded it because it could not deliver the packet. If the timer times out before a response comes in, the **traceroute** command prints an asterisk (*).

The **traceroute** command terminates when the destination responds, the maximum TTL is exceeded, or the user interrupts the trace with the escape sequence.

Examples

In the following example, the route for an IP address is displayed:

```
RP/0/RP0/CPU0:router# traceroute 10.233.233.233

Type escape sequence to abort.
Tracing the route to 10.233.233.233

 1  172.25.0.2 11 msec  2 msec  1 msec
 2  192.255.254.254 1 msec  *  2 msec
```

Using debug Commands

Debug commands are used to diagnose and resolve network problems. Use **debug** commands to troubleshoot specific problems or during troubleshooting sessions.

Use **debug** commands to turn on or off debugging for a specific service or subsystem. When debugging is turned on for a service, a debug message is generated each time the debugging code section is entered. The following sections provide information on debugging:

- Displaying a List of Debug Features, page 7-5
- Enabling Debugging for a Feature, page 7-5
- Disabling Debugging for a Service, page 7-6
- Displaying Debugging Status, page 7-5

Caution Debug commands can generate a very large amount of output and can render the system unusable. Use **debug** to troubleshoot specific problems or during specific troubleshooting sessions on systems that are not in production.

Displaying a List of Debug Features

To display a list of the available debug features, enter the debug mode and enter a **?** for on-screen help. For example:

```
RP/0/RP0/CPU0:router# debug
RP/0/RP0/CPU0:router(debug)# ?

  aaa                 AAA Authentication, Authorization and Accounting
  adjacency           Adjacency debug
  adjacency           platform AIB information
  aib                 AIB information
  alarm-logger        Turn on alarm debugging
  arm                 IP Address Repository Manager
  arp                 IP ARP transactions
  asic-errors         Debug ASIC erors
  asic-scan           Debug Asic Scan
--More--
```

Enabling Debugging for a Feature

To enable debugging for a feature, enter the **debug** command and then enable the feature for debugging. For example:

```
RP/0/RP0/CPU0:router# debug
RP/0/RP0/CPU0:router(debug)# aaa all
RP/0/RP0/CPU0:router(debug)# exit
```

You can also enter the complete command from EXEC mode, as shown in the following example:

```
RP/0/RP0/CPU0:router# debug aaa all
```

Displaying Debugging Status

Enter the **show debug** command to display the debugging features enabled for your terminal session. The terminal session is labeled *tty* and represents your connection to the router through a specific port, which might be the console port, auxiliary port, or Management Ethernet interface. In the following example, the command display indicates that debugging is enabled for two features (AAA and ipv4 io icmp) from a terminal session on the console port of RP1:

```
RP/0/RP0/CPU0:router# show debug

####  debug flags set from tty 'con0_RP1_CPU0'  ####
aaa all flag is ON
ipv4 io icmp flag is ON

RP/0/RP0/CPU0:router# no debug aaa all
RP/0/RP0/CPU0:router# show debug

####  debug flags set from tty 'con0_RP1_CPU0'  ####
ipv4 io icmp flag is ON
```

The preceding example is for a Cisco CRS-1. On a Cisco XR 12000 Series Router, the slot number of the tty ID is 0 or 1 instead of RP0 or RP1.

Enter the **show debug condition** command to display the conditional debugging status. For example:

```
RP/0/RP0/CPU0:router# show debug conditions

####  debug conditions set from tty 'con0_RP1_CPU0'  ####
interface  condition is ON for interface 'POS0/2/0/1'
```

Disabling Debugging for a Service

Use the **no** form of the **debug** command or the **undebug** command to turn off debugging for a service or subsystem.

In the following example, the **no debug** command disables debugging for the AAA feature:

```
RP/0/RP0/CPU0:router# no debug aaa all
RP/0/RP0/CPU0:router# show debug

####  debug flags set from tty 'con0_RP1_CPU0'  ####
ipv4 io icmp flag is ON
```

You can also turn off debugging from the undebug mode, as shown in the following example:

```
RP/0/RP0/CPU0:router# undebug
RP/0/RP0/CPU0:router(undebug)# aaa all
RP/0/RP0/CPU0:router(undebug)# exit
```

Disabling Debugging for All Services Started at the Active Terminal Session

Use the **undebug all** or **no debug all** command to turn off all debugging started by the active terminal session. For example, if you enter either of these commands while connected to the router through the console port on the active RP, all debug sessions started from that console port are disabled. In the following example, debugging for all services is disabled and then verified:

```
RP/0/RP0/CPU0:router# undebug all
RP/0/RP0/CPU0:router# show debug
No matching debug flags set
```

Disabling Debugging for All Services Started at All Terminal Sessions

Use the **undebug all all-tty** command to turn off debugging for all services that have been started from all terminal sessions. For example if you enter this command while connected to the router through the console port on the active RP, all debug sessions started from all ports are disabled. In the following example, debugging for all services and ports is disabled and then verified:

```
RP/0/0/CPU0:router# undebug all all-tty
RP/0/0/CPU0:router# show debug

No matching debug flags set
```

Understanding Processes and Threads

To achieve high availability and performance, the Cisco IOS XR software is built on a modular system of processes. Each process provides specific functionality for the system and runs in a protected memory space to ensure problems with one process cannot impact the entire system.

Multiple instances of a process can run on a single node, and multiple threads of execution can run on each process instance. Table 7-2 provides a summary of terms for processes and threads in the Cisco IOS XR software.

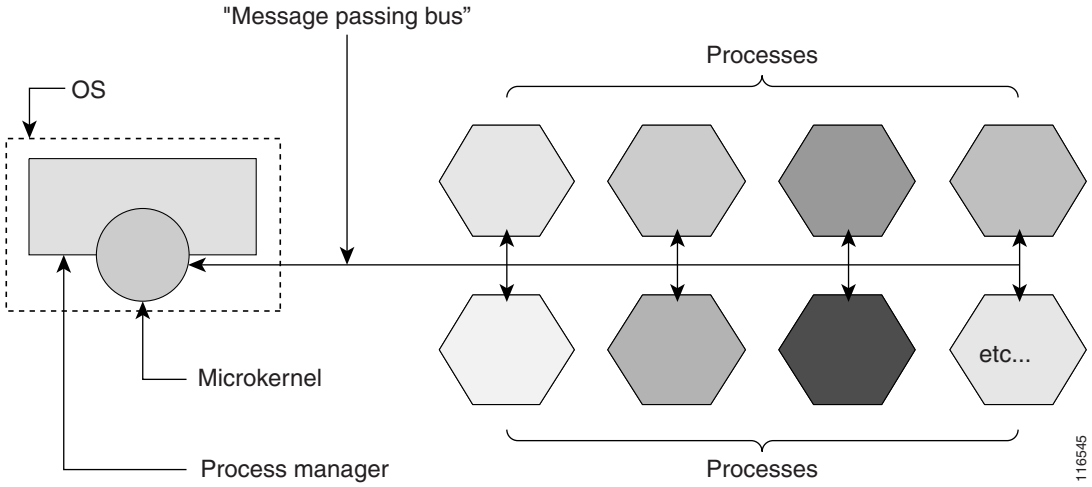
Table 7-2 Process and Thread Syntax and Descriptions

Term	CLI Syntax	Description
Process	• Process name • *executable-name*	A "process" is a group of threads that share a protected memory space. Processes run independently of other processes and can be individually started, restarted, or stopped. In the command-line interface (CLI) syntax, the process name (or *executable-name*) identifies all instances of a process on a node. Usage example: To change the core-dumping options for all instances of a process, specify the *executable-name* of the process.
Process *instance*	• JID • *job-ID*	Multiple instances of a process can run simultaneously on a node. In the CLI, the process instance is shown as the job ID. Usage example: To change the core-dumping options for only a single instance of a process, specify the job-ID of the process instance.
Thread	• TID • *thread-ID*	A "thread" is a unit of execution within a process. Multiple threads can run inside each instance of a process (known as "multithreading"). Each thread is assigned a thread ID number.

Under normal operating conditions, processes are managed automatically by the Cisco IOS XR software. Processes are started, stopped, or restarted as required by the running configuration of the router. In addition, processes are checkpointed to optimize performance during process restart and automatic switchover.

Figure 7-1 illustrates how the Cisco IOS XR software manages the operations of processes and acts as a message, passing "bus" to coordinate interactions between processes. In this way, processes can run independently, but still communicate and cooperate with other processes. If a process needs to be stopped or restarted, it affects only that process and related processes and threads.

Figure 7-1 Modular Process Architecture in Cisco IOS XR Software



Commands Used to Display Process and Thread Details

Table 7-3 describes some of the commands used to display information on the processes and threads running on a router.

For complete details on the commands and options related to process and thread management, see the *Cisco IOS XR System Management Command Reference*.

Table 7-3 Commands to Display Process and Thread Information

Command	Description
monitor processes	Displays the ten most active processes and the current CPU usage. The output from this command continually refreshes until quit (enter the **q**-key to quit).
monitor threads	Displays interactive, auto-updating process and thread statistics in a full-screen mode.
show processes	Displays information about active processes.
show processes abort	Displays process aborts.
show processes blocked	Displays details for reply, send, and mutually exclusive (mutex) blocked processes.
show processes boot	Displays process boot information.
show processes cpu	Displays CPU use for each process.
show processes dynamic	Displays process data for dynamically created processes.
show processes failover	Displays process automatic switchover information.
show processes log	Displays the process log.

Table 7-3 Commands to Display Process and Thread Information (continued)

Command	Description
show processes memory	Shows memory use for each process.
show processes startup	Shows process data for processes created at startup.

Examples

The following examples show the output and heading descriptions for commands commonly used to display information on processes and process memory usage:

- show processes Command, page 7-9
- show processes process-name Command, page 7-10
- show processes memory Command, page 7-11
- monitor processes Command, page 7-12

show processes Command

To display the running processes and information, such as the name, ID number, and state of the processes, enter the **show processes** command in EXEC mode. Table 7-4 describes the column heading output.

Table 7-4 Column Heading Descriptions for show processes Output

Output Heading	Description
JID	Job ID—In the CLI, the process instance is shown as the job ID (multiple instances of a process can run simultaneously on a node).
TID	Thread ID—A “thread” is a unit of execution within a process. Multiple threads can run inside each instance of a process (known as “multithreading”). Each thread is assigned a thread ID number.
LastCPU	CPU number on which the process was last running. The value is 0 or 1. For a single processor node, the value is 0.
Stack	Size of the memory stack of the process.
pri	Process priority.
state	Process state.
HR:MM:SS:MSEC	Time the process has run since starting.
NAME	Process name.

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

```
RP/0/RP0/CPU0:router# show processes

JID   TID LastCPU Stack pri state     HR:MM:SS:MSEC NAME
1       1    0       0K   0 Ready         1:57:41:0542 procnto-600-smp-cisco-instr
1       2    1       0K   0 Running       1:57:37:0022 procnto-600-smp-cisco-instr
1       3    1       0K  10 Receive       0:00:05:0723 procnto-600-smp-cisco-instr
1       4    1       0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1       5    0       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1       6    1       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1       7    0       0K  63 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1       8    0       0K  10 Receive       0:00:01:0885 procnto-600-smp-cisco-instr
1       9    1       0K  10 Receive       0:00:03:0416 procnto-600-smp-cisco-instr
1       10   1       0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1       11   1       0K  10 Receive       0:00:04:0861 procnto-600-smp-cisco-instr
1       15   0       0K  10 Receive       0:00:02:0020 procnto-600-smp-cisco-instr
1       18   1       0K  10 Receive       0:00:03:0278 procnto-600-smp-cisco-instr
1       20   1       0K  10 Receive       0:00:00:0732 procnto-600-smp-cisco-instr
1       21   1       0K  10 Receive       0:00:02:0692 procnto-600-smp-cisco-instr
1       22   0       0K  10 Running       0:00:03:0788 procnto-600-smp-cisco-instr
1       23   1       0K  10 Receive       0:00:11:0785 procnto-600-smp-cisco-instr
1       25   0       0K  10 Receive       0:00:04:0037 procnto-600-smp-cisco-instr
 --More--
```

show processes *process-name* Command

The **show processes** *process-name* command displays detailed information about a process. Table 7-5 describes the heading output.

Table 7-5 Heading Descriptions for show processes process-name Output

Output Heading	Description
JID	Job ID—This remains constant over process restarts. In the CLI, the process instance is shown as the job ID (multiple instances of a process can run simultaneously on a node).
PID	Process ID—This changes when process is restarted.
Executable path	Path for the process executable.
Instance	Instance of the process. More than one instance of a process may run at a given time (each instance may have more than one thread).
Version ID	API version.
Respawn	ON or OFF—Determines if this process restarts automatically in case of failure.
Respawn count	Number of times this process has been (re)started (that is, the first start makes this count 1).
Max. spawns per minute	Number of respawns not to be exceeded in 1 minute. If this number is exceeded, stop restarting.
Last started	Date and time the process was last started.
Process state	Current state of the process.
Started on config	Configuration command that started (or would start) this process.

Table 7-5 Heading Descriptions for show processes process-name Output

Output Heading	Description
core	Memory segments to include in a core file.
Max. core	Number of times to dump a core file—0 = infinity.

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

```
RP/0/RP0/CPU0:router# show processes ospf

                  Job Id: 261
                     PID: 139453
         Executable path: /hfr-rout-0.44.0/bin/ospf
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Wed Mar 17 07:46:26 2004
           Process state: Run
           Package state: Normal
       Started on config: cfg/gl/ipv4-ospf/proc/100/ord_a/routerid
                    core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                Mandatory: ON
                Placement: ON
             startup_path: /pkg/startup/ospf.startup
        Process cpu time: 0.410 user, 0.183 kernel, 0.593 total
JID    TID  LastCPU  Stack pri state        HR:MM:SS:MSEC NAME
261    1    0      40K  10 Receive       0:00:00:0397 ospf
261    2    1      40K  10 Receive       0:00:00:0003 ospf
261    3    0      40K  10 Receive       0:00:00:0007 ospf
261    4    1      40K  10 Condvar       0:00:00:0000 ospf
 --More--
```

show processes memory Command

The **show processes memory** command displays details of memory usage for a given process or all processes, as shown in the following example. Table 7-6 describes the column heading output.

Table 7-6 Column Heading Descriptions for show process memory Output

Output Heading	Description
JID	Job ID. In the CLI, the process instance is shown as the job ID (multiple instances of a process can run simultaneously on a node).
Text	Size of text region (process executable).
Data	Size of data region (initialized and uninitialized variables).
Stack	Size of process stack.
Dynamic	Size of dynamically allocated memory.
Process	Process name.

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

```
RP/0/RP0/CPU0:router# show processes memory

JID    Text     Data     Stack    Dynamic  Process
55     28672    4096     69632    17072128 eth_server
164    143360   4096     20480    13238272 hfr_fgid_server
317    167936   4096     45056    10526720 syslogd
122    512000   4096     77824    9797632  bgp
265    57344    4096     57344    5877760  parser_server
254    40960    4096     143360   3084288  netio
63     8192     4096     24576    2314240  nvram
314    4096     4096     36864    1699840  sysdb_svr_local
341    495616   4096     40960    1576960  wdsysmon
259    53248    4096     28672    1490944  nvgen_server
189    32768    4096     32768    1425408  hd_drv
69     77824    4096     110592   1421312  qnet
348    323584   4096     40960    1392640  ospf
347    323584   4096     40960    1392640  ospf
346    323584   4096     40960    1392640  ospf
345    323584   4096     40960    1392640  ospf
344    323584   4096     40960    1392640  ospf
261    323584   4096     40960    1392640  ospf
 --More--
```

monitor processes Command

This command shows the top ten processes of CPU usage. The display refreshes every 10 seconds. Table 7-7 describes the heading output. To change the **monitor processes** command to display parameters or terminate the display and return to the system prompt, see the interactive display characters described in Table 7-8.

Table 7-7 Heading Descriptions for monitor process Output

Output Heading	Description
JID	Job ID. In the CLI, the process instance is shown as the job ID (multiple instances of a process can run simultaneously on a node).
TIDS	Thread ID—A “thread” is a unit of execution within a process. Multiple threads can run inside each instance of a process (known as “multithreading”). Each thread is assigned a thread ID number.
Chans	Channels (client connections) to the server.
FDs	Number of files open.
Tmrs	Number of timers for the process.
MEM	Total memory of the process.
HH:MM:SS	Run time of process since last restart.
CPU	Percentage of CPU used by process thread.
NAME	Process name.

The following is sample output from the **monitor processes** command:

```
RP/0/RP0/CPU0:router# monitor processes

195 processes; 628 threads; 3300 channels, 4579 fds
CPU states: 47.6% idle, 1.2% user, 51.1% kernel
Memory: 2048M total, 1576M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU  NAME
        1   27  198     8    1     0    5:53:31 51.11% kernel
       52    5  215    44    5  228K    0:00:02  0.52% devc-conaux
      342    4  195    14    6    1M    0:00:08  0.34% wdsysmon
   495806    1    1    10    0  648K    0:00:00  0.16% ptop
      293    7   31    39   11  352K    0:00:09  0.07% shelfmgr
       55   11   24    14    5   16M    0:00:29  0.06% eth_server
      121    3   10     8    2  564K    0:00:05  0.02% bcm_process
      311    4    7    18    4  216K    0:00:02  0.01% sysdb_medusa_s
      138    4   14    40    5  240K    0:00:01  0.01% devc-vty
      265    5   31    19    4  204K    0:00:09  0.01% packet
```

To list the interactive commands, type **?** during the display. The options are described in Table 7-8.

Table 7-8 Interactive Display Commands for the monitor processes Command

Command	Description
?	Displays or prints the interactive commands.
q	Quits the monitor processes display and returns to the system prompt.
n	Changes the number of processes to be displayed.
d	Changes the delay interval between updates.
k	Kills a process.
l	Refreshes the screen.
t	Sorts the display by time (default).
m	Sorts the display by memory used.
c	Sorts the display by number of open channels.
f	Sorts the display by number of open files.

Commands Used to Manage Process and Threads

Table 7-9 describes the commands used to reset the options for a thread or to manually stop, start, and restart a process.

For complete details on the commands and options related to process and thread management, see the *Cisco IOS XR System Management Command Reference*.

Caution Manually stopping, starting, or restarting a process can seriously impact the operation of a router. Use these commands only under the direction of a technical support representative.

Table 7-9 Commands to Manage Processes

Command	Description
process {**kill** | **restart** | **start**}	Manually stops, starts, or restarts a process or process instance.
process mandatory {**on** | **off**}	Changes the mandatory setting for a process.
process mandatory reboot {**enable** | **disable**}	Changes how the system reacts to mandatory processes that go down.

Configuration Error Messages

The following sections contain information on configuration error messages:

Configuration Failures During a Commit Operation

A target configuration is added to the running configuration of a router when the **commit** command is entered. During this operation, the changes are automatically verified by the other components in the system. If successful, the configuration becomes part of the running configuration. If some configuration items fail, an error message is returned.

To display the configuration items that failed and see the cause of each failure, enter the **show configuration failed** command.

Note The **show configuration failed** command can be entered in either the EXEC mode or any configuration mode. In any mode, the configuration failures from the most recent **commit** operation are displayed.

In the following example, a configuration error occurs when an invalid commit operation is attempted:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup bgp
RP/0/RP0/CPU0:router(config-tg)# description this is a test of an invalid taskgroup
RP/0/RP0/CPU0:router(config-tg)# commit

% Failed to commit one or more configuration items. Please use 'show configurati
on failed' to view the errors
```

To display the configuration items that failed, including a description of the error, enter the **show configuration failed** command:

```
RP/0/RP0/CPU0:router(config-tg)# show configuration failed

!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
taskgroup bgp
!!% Usergroup/Taskgroup names cannot be taskid names
!
```

You can also display the failed configuration items without the error description by entering the **show configuration failed noerror** command:

```
RP/0/RP0/CPU0:router(config-tg)# show configuration failed noerror

!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
taskgroup bgp
```

Configuration Errors at Startup

Configuration errors that occurred during system startup can be displayed with the **show configuration failed startup** command. For example:

```
RP/0/RP0/CPU0:router# show configuration failed startup

!! CONFIGURATION FAILED DUE TO SYNTAX ERRORS
ntp
xml agent corba
http server
```

Memory Warnings in Configuration Sessions

The Cisco IOS XR software automatically monitors and manages the system resources in a router. Under normal operating conditions, memory problems should not occur.

When a low-memory issue does occur, it is often in the form of a low-memory warning during a configuration session. Low-memory conditions can be caused by multiple, large configurations being added to the router at a single time. Users can remove the source of a problem by removing configurations.

The following sections describe the commands used to display memory usage in a router and what to do if a low-memory warning appears:

Understanding Low-Memory Warnings in Configuration Sessions

The Cisco IOS XR software monitors memory usage in the Cisco CRS-1. If system memory becomes low, an error message is displayed when you attempt to enter configuration mode.

An "out-of-memory" error message is displayed during one of the following situations:

- When a user attempts to enter configuration mode.
- During a configuration session when the memory shortage occurs.
- When a user attempts to load a target configuration from a large file that results in a memory shortage.
- During a commit operation that results in the low-memory warning message. The commit operation is denied and only lr-root users can perform commit operations to remove configurations.

Caution Never ignore a low-memory warning. These warnings indicate a memory state that could affect system operations if not addressed.

"WARNING! MEMORY IS IN MINOR STATE"

If the system memory begins to run low, the following minor memory warning is displayed when you enter a new configuration mode.

```
WARNING! MEMORY IS IN MINOR STATE
```

Although users are allowed to enter configuration mode, they should immediately reduce memory usage using the tools described in the "Removing Configurations to Resolve Low-Memory Warnings" section on page 7-18.

Failure to take action can result in a worsening situation and eventual impact to router operations.

"ERROR! MEMORY IS IN SEVERE (or CRITICAL) STATE"

When the memory is in a severe or critical state, router operation and performance is likely to be affected. Regular users are not allowed to enter configuration mode. Only lr-root owners can enter configuration mode to free memory by removing configurations.

In some situations, the **commit** command is not allowed. Users with lr-root access can still use the **commit force** command to apply configurations that reduce memory usage. Reducing memory usage normally means removing configurations, but a user can also add configurations that reduce memory usage. For example, configuring the **shutdown** command on an interface could cause numerous routes to be purged from Border Gateway Protocol (BGP), the Routing Information Base (RIB), and Forwarding Information Base (FIB) configurations.

Caution The **commit force** command should be used only to apply configurations that reduce memory usage. Adding configurations that increase memory usage could result in serious loss of router operation.

Displaying System Memory Information

To display a high level summary of system memory, enter the **show memory summary** command. Table 7-9 describes the meaning of each heading.

```
RP/0/RP0/CPU0:router# show memory summary

Physical Memory: 2048M total
 Application Memory : 1787M (1509M available)
 Image: 132M (bootram: 132M)
 Reserved: 128M, IOMem: 0, flashfsys: 0
 Total shared window: 0
RP/0/RP1/CPU0:router#
```

To display general memory usage for the device as a whole and by process, enter the **show memory** command. Table 7-9 describes the meaning of each heading.

```
RP/0/RP0/CPU0:router# show memory

Physical Memory: 2048M total
 Application Memory : 1787M (1510M available)
 Image: 132M (bootram: 132M)
 Reserved: 128M, IOMem: 0, flashfsys: 0
 Total shared window: 0

kernel: jid 1
Address         Bytes           What
000d2000        12288           Program Stack
00112000        12288           Program Stack
Total Allocated Memory: 0
Total Shared Memory: 0

pkg/bin/wd-mbi: jid 72
Address         Bytes           What
4817f000        4096            Program Stack (pages not allocated)
48180000        516096          Program Stack (pages not allocated)
481fe000        8192            Program Stack
48200000        8192            Program Text
 --More--
```

Table 7-10 Heading Descriptions for show memory Command Output

Heading	Description
Physical Memory	Amount of physical memory installed on the device.
Application Memory	Memory available for the system to use (total memory minus image size, reserved, IOMem, and flashfsys).
Image	Size of the bootable image.
Reserved	Reserved for packet memory.
IOMem	IO memory—Currently used as a backup for packet memory.
flashfsys	Flash file system memory.
Process and JID	Process and job ID.
Address	Starting address in memory.
Bytes	Size of memory block.
What	Block description.

Removing Configurations to Resolve Low-Memory Warnings

To resolve most low-memory problems, you should remove the configurations from the router that are consuming the most memory. Often, memory problems occur when a large new configuration is added to the system. The following sections provide information to resolve low-memory issues:

- Clearing a Target Configuration, page 7-18
- Removing Committed Configurations to Free System Memory, page 7-18
- Rolling Back to a Previously Committed Configuration, page 7-20
- Clearing Configuration Sessions, page 7-20

Clearing a Target Configuration

A low-memory warning can occur when a large configuration file is loaded into a target configuration session. To remove the target configuration, enter the **clear** command to discard the changes. For example:

```
RP/0/RP0/CPU0:router(config)# clear
```

Caution Committing a target configuration that has caused a low-memory warning can make the system unstable. Clearing a target configuration is a preventive measure to not let the system go into a worse memory state due to additional configuration. In addition, all other active configuration sessions can be closed to minimize the churn.

Removing Committed Configurations to Free System Memory

You can reduce memory usage by removing configurations from the router, as shown in the following procedure:

Step 1 Enter the **show memory summary** command in EXEC mode to display the overall system memory:

```
RP/0/RP0/CPU0:router# show memory summary

Physical Memory: 2048M total
 Application Memory : 1787M (1511M available)
 Image: 132M (bootram: 132M)
 Reserved: 128M, IOMem: 0, flashfsys: 0
 Total shared window: 0
```

Step 2 Enter the **show configuration commit history** command in EXEC mode to see if a large configuration forced the router over the limit.

The output from this command does not show the details of the entries, but allows you to display a larger list of the commit events that occurred. To display the commitIDs to which you can roll back, use the **show configuration commit history** command.

```
RP/0/RP0/CPU0:router# show configuration commit history

SNo. Label/ID    User      Line        Client      Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~        ~~~~~~      ~~~~~~~~~~
1    1000000144  user      vty0        CLI         00:16:51 UTC Thu Dec 11 2003
2    1000000143  user      vty0        CLI         00:04:32 UTC Thu Dec 11 2003
3    1000000142  user      0.0.0.0     XMLAgent    21:58:36 UTC Wed Dec 10 2003
4    1000000141  user      0.0.0.0     XMLAgent    21:46:07 UTC Wed Dec 10 2003
5    1000000140  lab       con0_RP1_C  CLI         21:43:30 UTC Wed Dec 10 2003
6    1000000139  user      0.0.0.0     XMLAgent    21:40:13 UTC Wed Dec 10 2003
7    1000000138  user      0.0.0.0     XMLAgent    21:34:48 UTC Wed Dec 10 2003
8    1000000137  lab       con0_RP1_C  CLI         21:32:10 UTC Wed Dec 10 2003
9    1000000136  user      0.0.0.0     XMLAgent    21:30:13 UTC Wed Dec 10 2003
10   1000000135  lab       con0_RP1_C  CLI         19:45:04 UTC Wed Dec 10 2003
```

Step 3 Enter the **show configuration commit changes** command followed by a commitID number to display the configuration changes for a commit session (commitID). For example:

```
RP/0/RP0/CPU0:router# show configuration commit changes 1000000053

Building configuration...
interface preconfigure MgmtEth0/RP1/CPU0/0
 ipv4 address 10.8.50.10 255.255.0.0
 proxy-arp
!
route ipv4 172.255.254.254/32 12.8.0.1
end
```

Step 4 Remove the configuration using the appropriate configuration commands. In some situations, the **commit** command is not allowed. Users with lr-root access can still use the **commit force** command, but this command should be used only to remove configurations. The addition of new configurations seriously impacts router operation.

For more information, see the "Managing Configuration History and Rollback" section on page 3-3.

Rolling Back to a Previously Committed Configuration

You can roll back the system to a previous committed configuration, as shown in the following procedure:

Step 1 Enter the **show configuration commit list** command to display the commitIDs available for rollback:

```
RP/0/RP0/CPU0:router# show configuration commit list

SNo. Label/ID    User      Line        Client      Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~        ~~~~~~      ~~~~~~~~~~
1    1000000391  user_a    con0_33_1   CLI         19:29:18 UTC Wed Dec 10 2003
2    1000000390  user_a    con0_33_1   CLI         19:29:16 UTC Wed Dec 10 2003
3    1000000389  user_a    con0_33_1   CLI         19:29:15 UTC Wed Dec 10 2003
4    1000000388  user_a    con0_33_1   CLI         19:29:12 UTC Wed Dec 10 2003
5    1000000387  user_a    con0_33_1   CLI         19:26:16 UTC Wed Dec 10 2003
6    1000000386  user_a    con0_32_1   CLI         19:18:38 UTC Wed Dec 10 2003
7    1000000385  user_a    con0_33_1   CLI         19:14:09 UTC Wed Dec 10 2003
8    1000000384  user_a    con0_33_1   CLI         19:13:58 UTC Wed Dec 10 2003
9    1000000383  user_a    con0_33_1   CLI         19:13:33 UTC Wed Dec 10 2003
10   1000000382  user_a    con0_33_1   CLI         19:12:50 UTC Wed Dec 10 2003
```

Step 2 Enter the **show configuration rollback changes to** *commit-id* command to display the details of a specific commit ID:

```
RP/0/RP0/CPU0:router# show configuration rollback changes to 1000000373

Building configuration...
interface Loopback2
no description
no ipv4 address 10.0.5.1 255.0.0.0
```

Step 3 Roll back the configuration with the **rollback configuration to** *commitId* command:

```
RP/0/RP0/CPU0:router# rollback configuration to 1000000325

Configuration successfully rolled back to '1000000325'.
RP/0/RP0/CPU0:router#
```

For more information, see the "Managing Configuration History and Rollback" section on page 3-3.

Clearing Configuration Sessions

Active configuration sessions and their associated target configurations can consume system memory. Users with the appropriate access privileges can display the open configuration sessions of other users and terminate those sessions, if necessary (see Table 7-11).

Table 7-11 Session Commands

Command	Description
show configuration sessions	Displays the active configuration sessions.
clear configuration sessions *session-id*	Clears a configuration session.

In the following example, the open configuration sessions are displayed with the **show configuration sessions** command. The **clear configuration sessions** command is then used to clear a configuration session.

```
RP/0/RP0/CPU0:router# show configuration sessions

Session                   Line        User      Date                      Lock
00000211-002c409b-00000000  con0_RP1_CPU0  UNKNOWN   Mon Feb  2 01:02:09 2004

RP/0/RP0/CPU0:router# clear configuration sessions 00000211-002c409b-00000000

session ID '00000211-002cb09b-00000000' terminated
```

Contacting TAC for Additional Assistance

If you remove configurations and the low-memory condition remains, you may need to contact TAC for additional assistance. See the "Additional Sources for Information" section on page 1.

Interfaces Not Coming Up

The router interfaces are directly used in processing network traffic, so their status information is crucial to understanding how the device is functioning. This section contains information on the commands used to verify that the router interfaces are operational. The basic commands used in this process are summarized in Table 7-12.

Table 7-12 show interface Commands

Command	Description
show interface	Displays detailed information about all interfaces installed or configured on the device, whether or not they are operational.
show interface *type instance*	Specifies a particular interface, rather than displaying information for all interfaces, as in the following example: `show interface POS0/1/0/0`
show ipv4 interface	Displays basic, IP-related information for all available interfaces.
show ipv4 interface brief	Quickly displays the most critical information about the interfaces, including the interface status (up or down) and the IPv4 protocol status.

Verifying the System Interfaces

Perform the following steps to verify the system interfaces.

Step 1 Enter the **show platform** command to verify that all nodes are in the "IOS-XR RUN" state:

```
RP/0/RP0/CPU0:router# show platform

Node            Type            PLIM            State           Config State
--------------------------------------------------------------------------------
0/1/SP          MSC(SP)         N/A             IOS-XR RUN      PWR,NSHUT,MON
0/1/CPU0        MSC             16OC48-POS/DPT  IOS-XR RUN      PWR,NSHUT,MON
0/2/SP          MSC(SP)         N/A             IOS-XR RUN      PWR,NSHUT,MON
0/2/CPU0        MSC             16OC48-POS/DPT  IOS-XR RUN      PWR,NSHUT,MON
0/3/SP          MSC(SP)         N/A             IOS-XR RUN      PWR,NSHUT,MON
0/3/CPU0        MSC             16OC48-POS/DPT  IOS-XR RUN      PWR,NSHUT,MON
0/RP0/CPU0      RP(Active)      N/A             IOS-XR RUN      PWR,NSHUT,MON
0/RP1/CPU0      RP(Standby)     N/A             IOS-XR RUN      PWR,NSHUT,MON
0/SM0/SP        FC/S(SP)        N/A             IOS-XR RUN      PWR,NSHUT,MON
0/SM1/SP        FC/S(SP)        N/A             IOS-XR RUN      PWR,NSHUT,MON
```

Note Line cards in Cisco CRS-1s are called modular services cards (MSCs). The **show platform** command output is different for Cisco CRS-1s and Cisco XR 12000 Series Routers.

Step 2 Enter the **show ipv4 interface brief** command to verify IP address configuration and protocol status:

```
RP/0/RP0/CPU0:router# show ipv4 interface brief

Interface                      IP-Address      Status              Protocol
POS0/1/0/0                     unassigned      Shutdown            Down
POS0/1/0/1                     unassigned      Shutdown            Down
POS0/1/0/2                     unassigned      Shutdown            Down
POS0/1/0/3                     unassigned      Shutdown            Down
POS0/1/0/4                     unassigned      Shutdown            Down
POS0/1/0/5                     unassigned      Shutdown            Down
POS0/1/0/6                     unassigned      Shutdown            Down
POS0/1/0/7                     unassigned      Shutdown            Down
POS0/1/0/8                     unassigned      Shutdown            Down
POS0/1/0/9                     unassigned      Shutdown            Down
POS0/1/0/10                    unassigned      Shutdown            Down
POS0/1/0/11                    unassigned      Shutdown            Down
POS0/1/0/12                    unassigned      Shutdown            Down
POS0/1/0/13                    unassigned      Shutdown            Down
POS0/1/0/14                    unassigned      Shutdown            Down
POS0/1/0/15                    unassigned      Shutdown            Down
POS0/2/0/0                     10.10.1.101     Down                Down
POS0/2/0/1                     unassigned      Shutdown            Down
POS0/2/0/2                     unassigned      Shutdown            Down
POS0/2/0/3                     unassigned      Shutdown            Down
TenGigE0/3/0/0                 unassigned      Shutdown            Down
TenGigE0/3/0/2                 unassigned      Shutdown            Down
MgmtEth0/RP0/CPU0/0            unassigned      Shutdown            Down
```

Step 3 Configure the interfaces, as shown in the following examples.

Note You must enter the **commit** command to make the new configuration part of the active running configuration. If you end the configuration session, you are automatically prompted to commit the changes, as shown in the second example:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface pos0/2/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:router#

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface pos0/2/0/2
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.1.2 255.255.0.0
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router#
```

Step 4 Enter the **show ipv4 interface brief** command to verify that the interfaces are "Up" in the Status column:

```
RP/0/RP0/CPU0:router# show ipv4 interface brief

Interface                  IP-Address      Status              Protocol
POS0/1/0/0                 unassigned      Shutdown            Down
POS0/1/0/1                 unassigned      Shutdown            Down
POS0/1/0/2                 unassigned      Shutdown            Down
POS0/1/0/3                 unassigned      Shutdown            Down
POS0/1/0/4                 unassigned      Shutdown            Down
POS0/1/0/5                 unassigned      Shutdown            Down
POS0/1/0/6                 unassigned      Shutdown            Down
POS0/1/0/7                 unassigned      Shutdown            Down
POS0/1/0/8                 unassigned      Shutdown            Down
POS0/1/0/9                 unassigned      Shutdown            Down
POS0/1/0/10                unassigned      Shutdown            Down
POS0/1/0/11                unassigned      Shutdown            Down
POS0/1/0/12                unassigned      Shutdown            Down
POS0/1/0/13                unassigned      Shutdown            Down
POS0/1/0/14                unassigned      Shutdown            Down
POS0/1/0/15                unassigned      Shutdown            Down
POS0/2/0/0                 10.10.1.101     Up                  Up
POS0/2/0/1                 10.1.1.1        Up                  Up
POS0/2/0/3                 10.1.1.2        Shutdown            Down
POS0/2/0/3                 unassigned      Shutdown            Down
TenGigE0/3/0/0             unassigned      Shutdown            Down
TenGigE0/3/0/2             unassigned      Shutdown            Down
MgmtEth0/RP0/CPU0/0        unassigned      Shutdown            Down
```

Step 5 If the interface is in the "Shutdown/Down" state, as shown in the previous example, perform the following tasks:

a. Verify that the status of the interface is "Shutdown":

```
RP/0/RP0/CPU0:router# show running interface POS0/2/0/3

interface pos0/2/0/3
 shutdown
 keepalive disable
!
```

b. Bring the interface up with the following commands:

```
RP/0/RP0/CPU0:router(config)# controller pos 0/2/0/3
RP/0/RP0/CPU0:router(config-sonet)# no shutdown
RP/0/RP0/CPU0:router(config-sonet)# commit
RP/0/RP0/CPU0:router(config-sonet)# exit
RP/0/RP0/CPU0:router(config)# interface pos 0/2/0/3
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:router#
```

Step 6 If the interface state is still displayed as "Down," verify that the physical cable connections are correctly installed. The following message indicates that the interface has either a bad connection or no connection:

```
LC/0/0/1:Sep 29 15:31:12.921 : plim_4p_oc192[183]: %SONET-4-
ALARM : SONET0_1_1_0: SLOS
```

Step 7 Verify again that the interface is up by entering the **show ipv4 interface brief** command:

```
RP/0/RP0/CPU0:router# show ipv4 interface brief

Interface                  IP-Address      Status              Protocol
POS0/1/0/0                 unassigned      Shutdown            Down
POS0/1/0/1                 unassigned      Shutdown            Down
POS0/1/0/2                 unassigned      Shutdown            Down
POS0/1/0/3                 unassigned      Shutdown            Down
POS0/1/0/4                 unassigned      Shutdown            Down
POS0/1/0/5                 unassigned      Shutdown            Down
POS0/1/0/6                 unassigned      Shutdown            Down
POS0/1/0/7                 unassigned      Shutdown            Down
POS0/1/0/8                 unassigned      Shutdown            Down
POS0/1/0/9                 unassigned      Shutdown            Down
POS0/1/0/10                unassigned      Shutdown            Down
POS0/1/0/11                unassigned      Shutdown            Down
POS0/1/0/12                unassigned      Shutdown            Down
POS0/1/0/13                unassigned      Shutdown            Down
POS0/1/0/14                unassigned      Shutdown            Down
POS0/1/0/15                unassigned      Shutdown            Down
POS0/2/0/0                 10.10.1.101     Up                  Up
POS0/2/0/1                 10.1.1.1        Up                  Up
POS0/2/0/2                 10.1.1.2        Up                  Up
POS0/2/0/3                 unassigned      Shutdown            Down
TenGigE0/3/0/0             unassigned      Shutdown            Down
TenGigE0/3/0/2             unassigned      Shutdown            Down
MgmtEth0/RP0/CPU0/0        unassigned      Shutdown            Down
```

Step 8 Repeat these steps for every interface, until every interface shows both Status and Protocol as "Up."