



Cisco IOS XR Troubleshooting Guide for the Cisco CRS-1 Router

Cisco_IOS_XR_Software Release_3.9
October, 2010

Americas Headquarters

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

Text Part Number: OL-21483-02

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

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

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

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

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

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

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

Cisco IOS XR Troubleshooting Guide for the Cisco CRS-1 Router
© 2010 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface vii

Changes to This Document vii

Obtaining Documentation and Submitting a Service Request viii

CHAPTER 1

General Troubleshooting 1-1

Troubleshooting Techniques and Approaches 1-1

Documenting the Network 1-2

Verifying and Troubleshooting CLI Access 1-2

General CLI Access Information 1-2

User Access Privileges 1-3

CLI Access Through a Console Port 1-3

CLI Access Through a Terminal Server 1-3

CLI Access Through the Management Ethernet Interface 1-4

Basic Cisco IOS XR Verification and Troubleshooting Tools 1-7

man Command 1-7

describe Command 1-9

show platform Command 1-10

top Command 1-10

show context Command 1-11

show users Command 1-13

Verifying the System 1-13

Validating and Troubleshooting Cisco IOS XR Software Package Installation 1-27

Verifying the Software Version 1-27

Validating the Installation 1-30

install verify Command 1-30

show install active Command 1-32

show install committed Command 1-33

Validating and Troubleshooting Cisco IOS XR Software Configuration 1-35

Local and Global Configurations 1-35

Collecting Configuration Information 1-37

Verifying the Running Configuration 1-38

Using the show configuration failed Command 1-47

Startup Failed Configuration 1-47

Commit Configuration Failed 1-48

ASIC Errors	1-49
Trace Commands	1-56
Packets	1-57
Logging Archive for Harddisk	1-60
SNMP Polling Awareness of SystemOwner, LR Owner, MIB Location	1-60
Error File Locations and Data Collection Scripts	1-60
Error File Locations	1-61
harddisk:	1-61
Default disk location	1-62
Sysmgr Collection Scripts	1-62
Wdsysmon Collection Scripts	1-63
Shutdown Collection Scripts	1-66
ASIC error Collection Scripts	1-66
Monitoring	1-66
monitor interface Command	1-67
monitor controller Command	1-68
monitor processes Command	1-69
monitor threads Command	1-70
Gathering Information for Technical Support	1-70

CHAPTER 2

Troubleshooting Booting 2-73

Booting Tips	2-73
Verifying Successful Bootup	2-74
Verifying and Troubleshooting the Console Connection	2-74
Verifying and Troubleshooting Route Processor and Shelf Controller Cards	2-75
Troubleshooting RP and SC Cards Not Booting	2-75
Troubleshooting RP and SC Cards Resetting While Booting	2-78
Troubleshooting Blocked FCC Shelf Controller or LCC Route Process Minimum Boot Image Requests	2-79

CHAPTER 3

Troubleshooting Forwarding 3-81

Troubleshooting IPv4 CEF Information	3-81
Troubleshooting Adjacency Information	3-85
Troubleshooting Transient Traffic Drop in Forwarding	3-88
Troubleshooting Control Plane Information	3-90
Troubleshooting the Interface Manager	3-95
Interface Manager Control Process	3-96
Troubleshooting the Trace Logs for the Interface Manager	3-100

Troubleshooting the Client for the Interface Manager	3-101
Troubleshooting the Rules for the Interface Manager	3-101
Troubleshooting the Control Chain and Interface Information	3-103
Troubleshooting the Registrations for the Interface Manager	3-105
Troubleshooting the Interface Manager Distributor	3-106
Interface Manager Distributor Overview	3-106
Troubleshooting the Trace Logs for the Interface Manager Distributor	3-107
Troubleshooting the Clients for the Interface Manager Distributor	3-107
Troubleshooting the Interface Information for the Interface Manager Distributor	3-108
Troubleshooting the Global Registrations for the Interface Manager Distributor	3-109
Troubleshooting the Rules for the Interface Manager Distributor	3-110

CHAPTER 4
Troubleshooting Router Switch Fabric 4-111

CRS-1 Switch Fabric Overview	4-111
Understanding the Flags Field	4-113
Using the Online Diagnostics Tools	4-115
Verifying and Troubleshooting the Fabric Plane State	4-115
Verifying and Troubleshooting Up Fabric Planes	4-121
Troubleshooting Down Fabric Planes	4-126
Guidelines for Maintenance of Fabric Links	4-130

CHAPTER 5
Troubleshooting Interfaces 5-131

Verifying and Troubleshooting Configured Interfaces	5-131
Verifying and Troubleshooting Pluggable Optical Interfaces	5-139
Troubleshooting Common Issues with Bundle Interfaces	5-144

CHAPTER 6
Troubleshooting the Control Plane Ethernet Network 6-147

Cisco CRS-1 Control Plane Ethernet Network Overview	6-147
Using the Online Diagnostics Tools	6-148
Troubleshooting Booting the System Control Plane Ethernet Network	6-149
Examples	6-151
Troubleshooting the Multishelf System Router Topology	6-153
Examples	6-156
Troubleshooting the Catalyst Switch	6-158
Restrictions	6-158
Troubleshooting the CRS-1, 4-slot, 8-slot, or 16-slot System Router Topology	6-159
Examples	6-162

CHAPTER 7

Collecting System Information 7-165

- Capturing Logs 7-165
- Using ping and traceroute 7-166
- Using Debug Commands 7-166
- Using Diagnostic Commands 7-166
 - Online Diagnostics 7-166
 - Transient Condition when Standby RP Becomes Active 7-168
 - Offline Diagnostics—FDIAG RUNNING State 7-169
 - Additional Reference for Diagnostic Commands 7-169
- Commands Used to Display Process and Thread Details 7-169

CHAPTER 8

Process Monitoring and Troubleshooting 8-171

- System Manager 8-172
- Watchdog System Monitor 8-172
 - Deadlock detections 8-172
 - Hang detection 8-172
- Core Dumps 8-173
- follow Command 8-173
- show processes Commands 8-175
 - show processes boot Command 8-175
 - show processes startup Command 8-176
 - show processes failover Command 8-177
 - show processes blocked Command 8-178
 - Example: 8-179
- Redundancy and Process Restartability 8-179
- Process States 8-180
 - Synchronous Message Passing 8-181
 - Blocked Processes and Process States 8-181
- Process Monitoring 8-182
 - Process Monitoring Commands 8-183
- Monitoring CPU Usage and Using Syslog Messages 8-183
- Troubleshooting High CPU Utilization and Process Timeouts 8-185
 - General Guidelines for Troubleshooting CPU Utilization Problems 8-185
 - Using show process and top processes Commands 8-186
 - Troubleshooting a Process Block 8-188
 - Troubleshooting a Process Crash on Line Cards 8-192
 - Troubleshooting a Memory Leak 8-193
 - Troubleshooting a Hardware Failure 8-194

Troubleshooting SNMP Timeouts	8-194
Troubleshooting Communication Among Multiple Processes	8-194
Troubleshooting a Process Restart	8-195

CHAPTER 9**Troubleshooting Memory 9-197**

Watchdog System Monitor	9-197
Memory Monitoring	9-197
Configuring and Displaying Memory Thresholds	9-198
Setting Timeout for Persistent CPU Hogs	9-201
Memory Usage Analyzer	9-202
Troubleshooting Global Memory	9-202
Troubleshooting Process Memory	9-203
Identifying Process Memory Problems	9-203
Resolving Process Memory Problems	9-207

CHAPTER 10**Troubleshooting Upgrading and Downgrading Software 10-209**

Validating and Troubleshooting ROM Monitor Software Installation	10-209
Verifying and Troubleshooting the ROM Monitor Version	10-210
Troubleshooting Upgrading and Downgrading ROM Monitor Software on Cisco CRS-1 Routers	10-213
Troubleshooting Upgrading and Downgrading Cisco IOS XR Software Packages	10-214

CHAPTER 11**Troubleshooting the Statistics Infrastructure 11-215**

Debugging Statistics Infrastructure	11-215
debug statsd api Commands	11-215
debug statsd manager Commands	11-216
debug statsd server errors Command	11-216
Trace Commands for the Statistics Infrastructure	11-217
show statsd manager trace Command	11-217
show statsd server trace Command	11-218
show stats lib trace	11-218
Show Commands for the Statistics Infrastructure	11-218
show statsd manager info Command	11-219
show statsd collectors Command	11-220
show statsd registrations Command	11-222
show statsd requests Command	11-222
Diagnosing Problems with Statistics Values	11-223
Errors When Retrieving Data from the Statistics Manager EDM	11-224
Timeouts and Delays When Retrieving Data	11-224

Displaying Incorrect Rates for the show interfaces Command	11-225
Displaying Incorrect Data for the show Commands	11-226

CHAPTER 12

Multiprotocol Label Switching Checklist 12-229

Multiprotocol Label Switching Recommendations	12-229
Troubleshooting L3VPN MPLS Traffic Loss	12-230

CHAPTER 13

Troubleshooting Load Balancing 13-231

About Load Balancing with Cisco Express Forwarding	13-231
Load Balancing Function	13-231
Source Information for Load Balancing	13-231
Layer 2 Load Balancing	13-232
Terminology	13-233
Troubleshooting Layer 3 or Layer 4 Load Balancing	13-233
Verifying the Routing Table Entries for Parallel Links	13-234
Configuring Layer 4 Load Balancing	13-236
Verifying the CEF Database and Measuring Flows	13-236
Troubleshooting Layer 2 Load Balancing	13-237
Verifying the Bundle Status, IGP Route, and CEF Database	13-237
Configuring Layer 4 Load Balancing	13-238
Viewing the Expected Paths and Measuring the Flows	13-238
Configuration Examples for Troubleshooting Load Balancing	13-239
Troubleshooting Layer 3 or Layer 4 Load Balancing Example	13-240
Verifying Parallel Links in the Routing Table	13-240
Verifying IPv4/IPv6 Unicast Load Balancing	13-243
Troubleshooting Layer 2 Load Balancing Example	13-246
Verifying the Bundle Status	13-246
Checking the IGP Routing Table and CEF Database	13-247
Verifying Bundle Load Balancing	13-247

INDEX



Preface

This guide describes how to troubleshooting a router using the Cisco IOS XR software.

This preface contains the following sections:

- [Changes to This Document, page vii](#)
- [Obtaining Documentation and Submitting a Service Request, page viii](#)

Changes to This Document

[Table 1](#) lists the technical changes made to this document since it was first issued.

Table 1 *Changes to This Document*

Revision	Date	Change Summary
OL-21483-02	October, 2010	<p>Added information about the following topics:</p> <ul style="list-style-type: none">• Chapter 1—Added links to information on SNMP timeouts and the MIBs download site.• Chapter 1—Added a list of commands to use to gather information prior to contacting TAC.• Chapter 4—Added a drawing of the fabric architecture.• Chapter 4—Added information about using online diagnostics to help troubleshoot fabric.• Chapter 4—Added the CLI command show controllers fabric link health.• Chapter 4—Added information about using the correct fiber cleaning kit and procedure.• Chapter 4—Added a section with guidelines on maintenance of fabric links. <p>Continued, next page</p>

Table 1 **Changes to This Document (continued)**

Revision	Date	Change Summary
OL-21483-02	October, 2010	(Continued from previous page) <ul style="list-style-type: none"> • Chapter 5—Added a section on troubleshooting the line card optical interfaces (pluggable-optics). • Chapter 6—Added information on using online diagnostics to help troubleshoot the Ethernet control plane. • Chapter 7—Added CLI commands for displaying online diagnostics. • Chapter 7—Added a note about online diagnostics behavior after an RP failover. • Chapter 8—Added details on SNMP timeouts. • Chapter 8—Added information on normal and abnormal causes of high CPU utilization. • Chapter 10—Added clarifications regarding software and firmware upgrades.
OL-21483-01	December, 2009	Initial release of this document.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



CHAPTER 1

General Troubleshooting

This chapter describes general troubleshooting techniques you can use to troubleshoot routers using Cisco IOS XR software. This chapter includes the following sections:

- [Troubleshooting Techniques and Approaches, page 1-1](#)
- [Verifying and Troubleshooting CLI Access, page 1-2](#)
- [Basic Cisco IOS XR Verification and Troubleshooting Tools, page 1-7](#)
- [Verifying the System, page 1-13](#)
- [Validating and Troubleshooting Cisco IOS XR Software Package Installation, page 1-27](#)
- [Validating and Troubleshooting Cisco IOS XR Software Configuration, page 1-35](#)
- [ASIC Errors, page 1-49](#)
- [Trace Commands, page 1-56](#)
- [Packets, page 1-57](#)
- [Logging Archive for Harddisk, page 1-60](#)
- [SNMP Polling Awareness of SystemOwner, LR Owner, MIB Location, page 1-60](#)
- [Error File Locations and Data Collection Scripts, page 1-60](#)
- [Monitoring, page 1-66](#)
- [Gathering Information for Technical Support, page 1-70](#)

Troubleshooting Techniques and Approaches

The following techniques and approaches are recommended when troubleshooting using Cisco IOS XR software:

- Maintain current documentation about the network. See the [“Documenting the Network” section on page 1-2](#) for more information.
- Maintain current documentation about the system, including chassis numbers, serial numbers, installed cards, and location of chassis details.
- Maintain diagrams illustrating the connectivity of the router control plane Ethernet network.
- Capture and save the output of all commands. This information is useful when contacting Cisco Technical Support. For information on contacting Cisco Technical Support, see the [“Obtaining Documentation and Submitting a Service Request” section on page viii](#) in the [Preface](#).

- Have the output of the relevant **show tech-support** command captured and saved. The output from the **show tech-support** command provides a traditional dump of the configuration and **show** command outputs. For more commands used to collect system information, see [Chapter 7](#), “Collecting System Information.”

Documenting the Network

To be prepared to troubleshoot the network, maintain current documentation about the network, including the following:

- An up-to-date internetwork map that outlines the physical location of all the devices on the network and how they are connected, as well as a logical map of network addresses, network numbers, subnetworks, and so on
- A list of all network protocols implemented in your network; and for each of the protocols implemented, a list of the network numbers, subnetworks, zones, areas, and so on that are associated with them
- For multishelf systems, note the Layer 2 connections used to provide router control plane Ethernet network connectivity between racks (including the fabric card chassis [FCC])
- All points of contact to external networks
- The routing protocol for each external network connection
- The established baseline for your network, that is, the normal network behavior and performance at different times of the day so that you can compare any problems with a baseline
- Which device is the spanningtree root bridge for the system control plane Ethernet network

Verifying and Troubleshooting CLI Access

Ensure that the system has been booted. If the system has not booted, see *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on booting a router running Cisco IOS XR software. The following CLI access troubleshooting information is provided:

- [General CLI Access Information, page 1-2](#)
- [User Access Privileges, page 1-3](#)
- [CLI Access Through a Console Port, page 1-3](#)
- [CLI Access Through a Terminal Server, page 1-3](#)
- [CLI Access Through the Management Ethernet Interface, page 1-4](#)

General CLI Access Information

The following CLI access information applies to a console port, terminal server, and management Ethernet interface connections.

Once the terminal emulation software is started and you press **Enter**, a router prompt should appear. If no prompt appears, verify the physical connection to the console port and press **Enter** again. If the prompt still does not appear, contact Cisco Technical Support. See the “[Obtaining Documentation and Submitting a Service Request](#)” section on page viii in the [Preface](#) for Cisco Technical Support contact information.

If a prompt appears, indicating that the command-line interface (CLI) is accessible, but your login username and password are invalid, you are prevented from accessing the router. Verify that you have the correct username and password. If you have the correct username and password, but are locked out of the router, you may need to perform password recovery to access the system again. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for password recovery procedures.

User Access Privileges

When logging on to the router, use a username that is associated with a valid user group that has the authorization to execute the required commands.

- If you are troubleshooting all Secure Domain Routers (SDRs), the username must be associated with the root-system user group.
- If you are troubleshooting a single SDR, the username must be associated with the root-lr user group.

See *Cisco IOS XR System Security Command Reference for the Cisco CRS-1 Router* and *Cisco IOS XR System Security Configuration Guide for the Cisco CRS-1 Router* for information on users, usernames, and user groups.

CLI Access Through a Console Port

The first time a router is started, you must use a direct connection to the Console port to connect to the router and enter the initial configuration. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on connecting to the router through a console port. When you use a direct connection to the Console port, CLI commands are entered at a terminal or at a computer running terminal emulation software. A direct Console port connection is useful for entering initial configurations and performing some debugging tasks.

CLI Access Through a Terminal Server

A terminal server connection provides a way to access the Console port from a remote location. A terminal server connection is used when you need to perform tasks that require Console port access from a remote location.

Connecting to a router through a terminal server is similar to directly connecting through the Console port. For both connection types, the physical connection takes place through the Console port. The difference is that the terminal server connects directly to the Console port, and you must use a Telnet session to establish communications through the terminal server to the router.

If you are unable to access the CLI through a terminal server, perform the following procedure.

SUMMARY STEPS

1. Disable flow control (XON/XOFF) on the Terminal Server.
2. Disable local echo mode on the Terminal Server.
3. Verify the router name configured using the **hostname** command.
4. Check whether the port address is configured correctly.

5. Verify whether the address (interface) used for the reverse Telnet is up/up. The output of the **show interfaces brief** command provides this information. Cisco recommends you to use loopbacks because they are always up.
6. Ensure that you have the correct type of cabling. For example, you must not use a crossover cable to extend the length.
7. Establish a Telnet connection to the IP address port to test direct connectivity. You must Telnet from both an external device and the terminal server. For example, **telnet 172.21.1.1 2003**.
8. Ensure that you have the **transport input telnet** command under the line for the target device. The target device is the device that is connected to the terminal server.
9. Use a PC/dumb terminal to connect directly to the console of the target router. The target router is the device connected to the terminal server. This step helps you identify the presence of a port issue.
10. If you are disconnected, check timeouts. You can remove or adjust timeouts.

**Note**

Note: If you encounter authentication failures, remember that the terminal server performs the first authentication (if configured), while the device to which you try to connect performs the second authentication (if configured). Verify whether AAA is configured correctly on both the terminal server and the connecting device.

11. Contact Cisco Technical Support. See the “[Obtaining Documentation and Submitting a Service Request](#)” section on page viii in the [Preface](#) for Cisco Technical Support contact information.

CLI Access Through the Management Ethernet Interface

The Management Ethernet interface allows you to manage the router using a network connection. Before you can use the Management Ethernet interface, the interface must be configured. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on configuring the interface.

Once configured, the network connection takes place between client software on a workstation computer and a server process within the router. The type of client software you use depends on the server process you want to use. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on the client and server services supported by the Cisco IOS XR software.

If you are unable to access the CLI through a management Ethernet interface, perform the following procedure.

SUMMARY STEPS

1. **show interface MgmtEth** *interface-instance*
2. **show arp MgmtEth** *interface-instance*
3. **show ipv4 interface** *type instance*
4. **ping**
5. Contact Cisco Technical Support if the problem is not resolved

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>show interfaces MgmtEth <i>interface-instance</i></p> <p>Example: RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/rp0/cpu0/0</p>	<p>Displays statistics for all interfaces configured on the router.</p> <p>Check the following:</p> <ul style="list-style-type: none"> • MgmtEth interface is up • Line protocol (state of the Layer 2 line protocol) is up • Number of input and output errors <p>If an interface is down, use the no shutdown command to enable the interface.</p> <p>If the interface is up and the input and output errors are within an acceptable range, proceed to Step 2.</p> <p>If input or output errors are not within an acceptable range, the management Ethernet interface is not enabled when the no shutdown command is used, or the line protocol is down, see Chapter 5, “Troubleshooting Interfaces,” for detailed information on troubleshooting interfaces.</p>
Step 2	<p>show arp MgmtEth <i>interface-instance</i></p> <p>Example: RP/0/RP0/CPU0:router# show arp MgmtEth 0/rp0/cpu0/0</p>	<p>Displays the Address Resolution Protocol (ARP) table for the management Ethernet interface.</p> <p>Ensure that the expected ARP entries exist for the management Ethernet interface.</p> <p>If the expected ARP entries do not exist, verify the physical layer Ethernet interface connectivity. Use the show arp trace command to display the ARP entries in the buffer.</p> <p>See the Chapter 5, “Troubleshooting Interfaces,” for more information on troubleshooting interfaces.</p> <p>If the expected ARP entries exist, proceed to Step 3.</p>
Step 3	<p>show ipv4 interface <i>type instance</i></p> <p>Example: RP/0/RP0/CPU0:router# show ipv4 interface MgmtEth 0/rp0/cpu0/0</p>	<p>Displays the usability status of interfaces configured for IPv4.</p> <p>If the status of the interface is not as expected, see Chapter 5, “Troubleshooting Interfaces,” for more information on troubleshooting interfaces.</p> <p>If the is in the expected state, proceed to Step 4</p>
Step 4	<p>ping</p> <p>Example: RP/0/RP0/CPU0:router# ping</p>	<p>Checks host reachability and network connectivity on the IP network.</p> <p>If no problems are detected, proceed to Step 5.</p>
Step 5	Contact Cisco Technical Support.	<p>If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the “Obtaining Documentation and Submitting a Service Request” section on page viii in the Preface.</p>

Examples

The output from the **show interfaces MgmtEth** command displays the status of the management Ethernet interface. For example, in the following output the management Ethernet interface is up. MgmtEth0/RP1/CPU0/0 is up indicates that the interface hardware is currently active and line protocol is up indicates that the keep a lives are successful. There are 42 input errors and 0 output errors.

```
RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/rp1/cpu0/0

MgmtEth0/RP1/CPU0/0 is up, line protocol is up
  Hardware is Management Ethernet, address is 0011.93ef.e8fe (bia 0011.93ef.e8fe
)
  Description: Connected to Lab LAN
  Internet address is 172.29.52.71/24
  MTU 1514 bytes, BW 100000 Kbit
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA,
  Half-duplex, 100Mb/s, link type is autonegotiation
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    122444 packets input, 7450512 bytes, 45 total input drops
    0 drops for unrecognized upper-level protocol
  Received 98306 broadcast packets, 0 multicast packets
    0 runts, 0 giants, 0 throttles, 0 parity
  42 input errors, 37 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  26741 packets output, 5100214 bytes, 0 total output drops
  Output 48 broadcast packets, 0 multicast packets
  0 output errors, 0 underruns, 0 applique, 0 resets
  0 output buffer failures, 0 output buffers swapped out
  10 carrier transitions
```

The output from the **show arp MgmtEth 0/rp1/cpu0/0** command displays the ARP table for the management Ethernet interface.

```
RP/0/RP0/CPU0:router# show arp MgmtEth 0/rp1/cpu0/0

-----
0/RP1/CPU0
-----

```

Address	Age	Hardware Addr	State	Type	Interface
10.86.154.82	01:32:10	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.128	03:55:55	0013.c4cb.a200	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
10.21.82.85	00:09:17	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.20.212.227	03:51:56	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.18.196.200	02:32:14	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.201	00:07:54	0010.7b3c.6847	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.200	00:09:56	0010.7b3c.689f	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.19.16.196	00:11:19	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.1	00:10:41	0030.f2f2.1038	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.121	01:41:03	0012.da0b.97ff	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.120	01:41:20	0004.2892.c7ff	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.127	03:55:56	0013.c4cb.a200	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.71	-	0011.93ef.e8fe	Interface	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.70	00:26:29	0011.93ef.e8ea	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.73	01:42:04	0014.a9bc.6600	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.72	-	0011.93ef.e8fe	Interface	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.75	00:11:14	0011.93ef.e8e2	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.77	00:11:14	0011.93ef.e8e2	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.76	02:12:54	0011.93ef.e8e6	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0
172.29.52.78	01:42:02	0014.a8de.6700	Dynamic	ARPA	MgmtEth0/RP1/CPU0/0


```

172.23.105.135 03:15:34 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
161.44.192.216 02:19:31 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.22.45.18 01:00:50 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.22.45.17 01:09:39 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.23.93.112 00:22:35 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.22.58.32 00:22:32 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.27.90.107 02:58:54 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
172.28.54.111 02:45:53 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
171.71.180.204 02:12:47 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
171.71.180.203 02:12:55 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0
171.71.180.216 02:12:48 0030.f2f2.1038 Dynamic ARPA MgmtEth0/RP1/CPU0/0

```

Use the output from the **show arp MgmtEth 0/rp1/cpu0/0** command to verify that there are dynamic ARP addresses in the table and that ARP is functioning over the interface. The output shows that ARP is functioning over the management Ethernet interface 0/RP1/CPU0.

The **ping** command checks to see if the neighbor is reachable.

```
RP/0/RP0/CPU0:router# ping 172.16.0.1 count 10 source mgmteth0/rp0/cpu0/0
```

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 172.16.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/2/4 ms

Basic Cisco IOS XR Verification and Troubleshooting Tools

The following commands are used to collect information to aid in verifying the system and troubleshooting problems:

- [man Command, page 1-7](#)
- [describe Command, page 1-9](#)
- [show platform Command, page 1-10](#)
- [top Command, page 1-10](#)
- [show context Command, page 1-11](#)
- [show users Command, page 1-13](#)

man Command

The **man** command provides online help for standard Cisco IOS XR command-line interface (CLI) commands using manual (man) pages. The command is used to display the manual pages for a specific command on the basis of the command name, a feature, or a keyword. Each man page contains the command name, syntax, command mode, usage, examples, and related commands.



Note

The Cisco IOS XR Documentation Package - Man pages for Cisco IOS XR CLI commands must be loaded in order to run the man command.

The following example shows the output from the **man command show users** command.

```
RP/0/RP0/CPU0:router# man command show users
```

```
COMMAND
```

```
show users
```

DESCRIPTION

To display information about the active lines on the router, use the show users command in EXEC mode.

```
show users
```

SYNTAX DESCRIPTION

This command has no arguments or keywords.

DEFAULTS

No default behavior or values

COMMAND MODES

EXEC

COMMAND HISTORY

Release

Modification

Release 2.0

This command was introduced on the Cisco CRS-1.

Release 3.0

No modification.

Release 3.2

This command was first supported on the Cisco XR 12000 Series Router.

USAGE GUIDELINES

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, see the Configuring AAA Services on Cisco IOS XR Software module of the Cisco IOS XR System Security Configuration Guide.

Use the show users command to display the line number, connection name, idle time, hosts, and terminal location. An asterisk (*) indicates the current terminal session.

EXAMPLES

The following is sample output identifying an active vty terminal session:

```
* * * * * START OF LISTING * * * * *
```

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

```
* * * * * END OF LISTING * * * * *
```

```
Line User Service Conns Idle Location
con0_RP0_CPU0 cisco hardware 0 18:33:48
vty0 cisco telnet 0 00:30:36 10.33.54.132
* vty1 cisco telnet 0 00:00:00 10.33.54.132
```

```
* * * * * END OF LISTING * * * * *
```

Table 89 describes the significant fields shown in the display.

Table^B^`89 show users Field Descriptions^B^`

Field	Description
Line	All current connections. An asterisk (*) indicates the active connection.
User	Username of the user logged into the line.
Service	Physical or remote login service used.
Conns	Number of outgoing connections.
Idle	Interval (in hours:minutes:seconds) since last keystroke.
Location	IP address of remote login host. For local (physical) terminal connections, this field is blank.

RELATED COMMANDS

Command	Description
show line	Displays the parameters of a terminal line.

describe Command

The **describe** command provides package, component, and task ID information for a specific command. You must be in the appropriate configuration mode for the specific command. For example, to display the package, component, and task ID information for the **router bgp 1** command, you must be in global configuration mode.

The following example shows the output from the **describe router bgp 1** command.

```
RP/0/RP0/CPU0:router(config)# describe router bgp 1
```

```
Package:
  hfr-rout
    hfr-rout V3.3.0 Routing Package
    Vendor  : Cisco Systems
    Desc    : Routing Package
    Build   : Built on Tue Jan 31 10:56:38 UTC 2006
    Source  : By edde-bld1 in /files/3.3.0/hfr/workspace fo8
    Card(s) : RP, DRP, DRPSC
```

```
Component:
  ipv4-bgp V[fwd-33/53]  IPv4 Border Gateway Protocol (BGP)
```

User needs ALL of the following taskids:

```
bgp (READ WRITE)
```

show platform Command

The **show platform** command displays a high level overview of the entire physical system. Use the **show platform** command in administration mode to display a summary of the nodes in the system, including node type and status.



Note

The **show platform** command in EXEC mode displays a high level overview of the specific secure domain router (SDR).

The following example shows the output from the **show platform** command in administration mode.

```
RP/0/RP0/CPU0:router(admin)# 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	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/1/0	MSC (SPA)	4XOC3-POS	OK	PWR, NSHUT, MON
0/1/5	MSC (SPA)	8X1GE	OK	PWR, NSHUT, MON
0/6/SP	MSC (SP)	N/A	FDIAG RUNNING	PWR, NSHUT, MON
0/6/CPU0	MSC	Jacket Card	FDIAG RUNNING	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
0/SM2/SP	FC/S (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM3/SP	FC/S (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON

top Command

The **top** command is used to monitor CPU usage on the system through interactive process statistics.

The following example show the output from the **top** command.

```
RP/0/RP0/CPU0:router# top
```

```
Computing times...
```

```
224 processes; 803 threads;
```

```
CPU states: 93.6% idle, 3.6% user, 2.7% kernel
```

```
Memory: 4096M total, 3504M avail, page size 4K
```

JID	TID	PRI	STATE	HH:MM:SS	CPU	COMMAND
65555	2	10	Rcv	4:59:34	1.51%	devb-ata
1	6	10	Run	0:15:01	1.20%	procnto-600-smp-cisco-instr
1	13	10	Rcv	0:39:58	1.03%	procnto-600-smp-cisco-instr
57	5	10	Rcv	0:27:47	0.53%	dllmgr
1	28	10	Rcv	0:34:59	0.47%	procnto-600-smp-cisco-instr
65756	1	10	Rply	0:00:00	0.20%	top
65555	7	10	Rcv	0:00:00	0.10%	devb-ata
59	7	55	Rcv	0:22:50	0.09%	eth_server
59	9	10	Rcv	0:05:13	0.09%	eth_server
319	5	10	Rcv	0:15:38	0.07%	shelfmgr

Press 'q' to exit the command.

show context Command

The **show context** command displays core dump context information for the last ten core dumps. The command output is used for post-analysis in the debugging of processes (determine if any process crashes have occurred).

If there are no crashed processes, the **show context** command displays no output for each node. The following example shows the output of the **show context** command with no crashed processes.

```
RP/0/RP1/CPU0:router# show context
```

```
node:      node0_1_CPU0
```

```
-----
```

```
node:      node0_6_CPU0
```

```
-----
```

```
node:      node0_RP0_CPU0
```

```
-----
```

```
node:      node0_RP1_CPU0
```

```
-----
```

The following example shows the output from the **show context** command where there is a crashed process.

```
RP/0/RP1/CPU0:router# show context
```

```
node:      node0_1_CPU0
```

```
-----
```

```
Crashed pid = 61524 (pkg/bin/tcam_mgr)
```

```
Crashed tid = 1
```

```
Crash time: Wed Apr 05, 2006: 18:27:26
```

```
Core for process at harddisk:/dumper/first.tcam_mgr.abort.node0_1_CPU0.ppc.Z
```

Stack Trace

```
#0 0xfc1d3fa0
#1 0xfc1c6340
#2 0xfc1c5364
#3 0xfc1c542c
#4 0x48210930
#5 0x482110b8
#6 0x48212ba4
#7 0x48203dd8
#8 0x4820c61c
#9 0xfc1557ec
#10 0xfc15573c
#11 0xfc152fb8
#12 0x4820d140
```

Registers info

	r0	r1	r2	r3
R0	00000000	481ff7b0	4824a55c	00000000
	r4	r5	r6	r7
R4	0000f054	00000001	00000006	00000000
	r8	r9	r10	r11
R8	00000000	fc220000	481fffc0	00000000
	r12	r13	r14	r15
R12	4823be90	4824a4a0	48230000	00000000
	r16	r17	r18	r19
R16	00000048	00000001	00000019	48256520
	r20	r21	r22	r23
R20	00000000	00000000	00000003	00000045

```

      r24      r25      r26      r27
R24  00000003  00000000  00000003  4825dc34
      r28      r29      r30      r31
R28  00000006  0000f054  48254064  481ff810
      cnt      lr      msr      pc
R32  00000000  fc1c6340  0000d932  fcd1d3fa0
      cnd      xer
R36  28004024  00000008

      DLL Info
DLL path      Text addr.  Text size  Data addr.  Data size  Version
/hfr-os-3.3.90/lib/libinfra.dll  0xfc142000  0x00034200  0xfc1343b8  0x00000bbc
0
/lib/libc.dll  0xfc1a8000  0x00079dd8  0xfc222000  0x00002000  0

      Crash Package Infomation
Package: hfr-mgbl, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-mcast, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-mpis, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-rout, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-k9sec, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-lc, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-fwgdg, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-admin, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-base, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-os-mbi, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8

node:      node0_6_CPU0
-----

node:      node0_RP0_CPU0
-----

node:      node0_RP1_CPU0
-----

```

Use the **show context** command to locate the core dump file path. For example, the core dump file path shown in the command output is: `harddisk:/dumper/first.tcam_mgr.abort.node0_1_CPU0.ppc.Z`. The command output shows a crashed on a node. The process is `pkg/bin/tcam_mgr`.

Collect the following information and sent it to Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the Preface.

- `ppc.Z` file—This file contains the binary core dump information. Use the path listed in the command output to copy the contents of the `ppc.Z` file. The path shown in the command output is: `harddisk:/dumper/first.tcam_mgr.abort.node0_1_CPU0.ppc.Z`
- `ppc.txt` file—This file contains content on the core dump similar to the **show context** command output. Use the path listed in the command output to copy the contents of the `ppc.txt` file. The path shown in the command output is: `harddisk:/dumper/first.tcam_mgr.abort.node0_1_CPU0.ppc.txt`
- Collect the **show version** or **show install active** command output.

show users Command

The **show users** command displays information on active lines on the router including the line number, user, service, number of connections, idle time, and remote terminal location. An asterisk (*) indicates the current terminal session.

The following example shows the output from the **show users** command.

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

	Line	User	Service	Conns	Idle	Location
*	vty0	User_A	telnet	0	00:00:00	161.44.1925
	vty1	User-B	telnet	0	00:00:03	161.44.1929

Verifying the System

To verify the general status and state of a router using Cisco IOS XR software, perform the following procedure.

SUMMARY STEPS

1. **admin**
2. **show platform** *[node-id]*
3. **show version**
4. **show running-config**
5. **show logging**
6. **show environment**
7. **show context**
8. **exit**
9. **show context**
10. **show memory summary detail location all**
11. **show memory heap summary** *{job-id | all}*
12. **top processes**
13. **show running-config**
14. **show system verify start**
show system verify report
15. **show {ipv4 | ipv6} interface brief**

DETAILED STEPS

	Command or Action	Purpose
Step 1	admin Example: RP/0/RP0/CPU0:router# admin	Enters administration mode.
Step 2	show platform [<i>node-id</i>] Example: RP/0/RP0/CPU0:router(admin)# show platform	Displays information about the status of cards and modules installed in the router. <ul style="list-style-type: none"> Some cards support a CPU module and service processor (SP) module. Other cards support only a single module. A card module is also called a <i>node</i>. When all nodes are working properly, the status of each node displayed in the State column is IOS-XR RUN. Type the show platform node-id command to display information for a specific node. Replace <i>node-id</i> with a node name from the show platform command Node column.
Step 3	show version Example: RP/0/RP0/CPU0:router(admin)# show version	Displays information about the router, including image names, uptime, and other system information. Verify that the expected software version and images are installed.
Step 4	show running-config Example: RP/0/RP0/CPU0:router(admin)# show running-config	Displays hardware module power status, secure domain router (SDR) configuration, and fabric configuration. The output also displays the users defined in administration mode with root-system access. For Cisco CRS-1 Multishelf Systems, it displays the rack numbers and serial numbers for the nodes in the currently running administration configuration. Verify that the rack numbers and serial numbers for the nodes in the current running configuration are what is expected. The expected rack numbers and serial numbers should be listed in the current system documentation. See the “Troubleshooting Techniques and Approaches” section on page 1-1. Also verify that the hardware module power status is as expected and the SDR and fabric configurations are as expected.
Step 5	show logging Example: RP/0/RP0/CPU0:router(admin)# show logging	Displays all syslog messages stored in the buffer. The command output displays the device operation history from a system perspective. Analyze the logged events and their order of happening. Check for anything out of the ordinary such as errors, tracebacks, or crashes. Also check for any Severity 1 or Severity 2 errors.

	Command or Action	Purpose
Step 6	show environment Example: RP/0/RP0/CPU0:router(admin)# show environment	Displays display environmental monitor parameters for the system. Verify that the parameters are as expected.
Step 7	show context Example: RP/0/RP0/CPU0:router(admin)# show context	Displays core dump context information on fabric cards, alarm modules, fan controllers, and service processors (system-owned cards). See the “show context Command” section on page 1-11 for more information on the show context command output.
Step 8	exit Example: RP/0/RP0/CPU0:router(admin)# exit	Exits administration mode.
Step 9	show context Example: RP/0/RP0/CPU0:router# show context	Displays core dump context information on CPUs responsible for routing and Cisco Express Forwarding (CEF). See the “show context Command” section on page 1-11 for more information on the show context command output.
Step 10	show memory summary detail location all Example: RP/0/RP0/CPU0:router# show memory summary detail location all	Displays information about the memory available on the router after the system image decompresses and loads. Verify that the expected memory is available or installed. Ensure that all memory regions have adequate free space available.
Step 11	show memory heap summary {job-id all} Example: RP/0/RP0/CPU0:router# show memory heap summary all	Displays a summary of the information about the heap space. The output displays each process and the amount of memory allocated for each process. Verify if there are any processes using a large amount of memory.
Step 12	top processes Example: RP/0/RP0/CPU0:router# top processes	To get a live update of process resource consumption, use the top processes command and press ‘M’ to sort by memory usage. Verify that the resource consumption is as expected.
Step 13	show running-config Example: RP/0/RP0/CPU0:router# show running-config	Displays the contents of the currently running configuration. Verify that the contents of the current running configuration are what is expected.

Command or Action	Purpose
Step 14 <code>show system verify start</code> <code>show system verify report</code> Example: RP/0/RP0/CPU0:router# <code>show system verify start</code> RP/0/RP0/CPU0:router# <code>show system verify report</code>	A two-step command that produces system reports. <ul style="list-style-type: none"> • show system verify start—Starts the system verify process (creates the initial baseline file) • show system verify report—Generates a report for the system verification process (report of the current status) <p>The output of the show system verify report command provides a comparison of the system at the time of the show system verify start snapshot and the show system verify report snapshot. The output provides a sanity check of the system provided the show system verify start system snapshot was taken when the system was healthy or before an event.</p> <p>Verify that the system parameters are as expected.</p>
Step 15 <code>show {ipv4 ipv6} interface brief</code> Example: RP/0/RP0/CPU0:router# <code>show ipv4 interface brief</code>	Displays the usability status of interfaces. Verify that all expected interfaces are listed, that they have the correct assigned address, and that they are in the expected states.

Examples

The output from the **show platform** command indicates that all expected nodes are in the run state. If all nodes in the system are active, the cards should be in the IOS XR RUN and the SPAs should be in the OK state.

RP/0/RP0/CPU0:router(admin)# **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	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/1/0	MSC (SPA)	4XOC3-POS	OK	PWR, NSHUT, MON
0/1/5	MSC (SPA)	8X1GE	OK	PWR, NSHUT, MON
0/6/SP	MSC (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/6/CPU0	MSC	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/6/0	MSC (SPA)	4XOC3-POS	OK	PWR, NSHUT, MON
0/6/4	MSC (SPA)	8XOC3/OC12-POS	OK	PWR, NSHUT, MON
0/6/5	MSC (SPA)	8X1GE	OK	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
0/SM2/SP	FC/S (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM3/SP	FC/S (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON

The example output shows that all expected nodes are in the run state.

The output from the **show version** command indicates the version of software being run on the nodes and from which location (disk or network). Check that the expected software version and images are installed.

```

RP/0/RP0/CPU0:router(admin)# show version

Cisco IOS XR Software, Version 3.3.0
Copyright (c) 2006 by cisco Systems, Inc.

ROM: System Bootstrap, Version 1.32(20050525:193559) [CRS-1 ROMMON],

CRS-8_X1 uptime is 1 week, 6 days, 23 hours, 5 minutes
System image file is "disk0:hfr-os-mbi-3.3.0/mbihfr-rp.vm"

cisco CRS-8/S (7457) processor with 4194304K bytes of memory.
7457 processor at 1197Mhz, Revision 1.2

16 Packet over SONET/SDH network interface(s)
16 SONET/SDH Port controller(s)
2 Ethernet/IEEE 802.3 interface(s)
16 GigabitEthernet/IEEE 802.3 interface(s)
2043k bytes of non-volatile configuration memory.
38079M bytes of hard disk.
1000592k bytes of ATA PCMCIA card at disk 0 (Sector size 512 bytes).
1000640k bytes of ATA PCMCIA card at disk 1 (Sector size 512 bytes).

Package active on node 0/1/SP:
hfr-diags, V 3.3.0, Cisco Systems, at disk0:hfr-diags-3.3.0
  Built on Tue Jan 31 13:32:40 UTC 2006
  By edde-bld1 in /files/3.3.0/workspace for c2.95.3-p8

hfr-admin, V 3.3.0, Cisco Systems, at disk0:hfr-admin-3.3.0
  Built on Tue Jan 31 10:02:02 UTC 2006
  By edde-bld1 in /files/3.3.0/workspace for c2.95.3-p8
.
.
.

```

The example output shows that the Cisco IOS XR software version is 3.3.0 and that the installed pie versions are also 3.3.0.

The output from the **show running-config** command displays hardware module power status, secure domain router (SDR) configuration, fabric configuration, and rack numbers and serial numbers. The rack and serial numbers are displayed for Cisco CRS-1 Multishelf Systems only.

```

RP/0/RP0/CPU0:router(admin)# show running-config

Building configuration...
username user_A
  secret 5 $1$SopS$KK2gkdTQYDTKMbWMILZ5P1
  group root-system
!
dsc serial TBA08260159 rack 0
dsc serial TBA08440024 rack 1
dsc serial TBC0820052000000 rack 240
controllers fabric plane 0 topology single-module location F0/SM0/FM
controllers fabric plane 1 topology single-module location F0/SM1/FM
controllers fabric plane 4 topology single-module location F0/SM4/FM
controllers fabric plane 5 topology single-module location F0/SM5/FM
end

```

The example output shows the contents of the current running configuration for a Cisco CRS-1 Multishelf System.

The output from the **show logging** command displays the contents of the logging buffer. The output displays details on syslog historical events. Analyze the logged events and their order of happening. Check for anything out of the ordinary such as errors, tracebacks, or crashes. Also check for any Severity 1 or Severity 2 errors.

```
RP/0/RP0/CPU0:router(admin)# show logging

Syslog logging: enabled (63 messages dropped, 0 flushes, 0 overruns)
  Console logging: level informational, 16647 messages logged
  Monitor logging: level debugging, 0 messages logged
  Trap logging: level informational, 0 messages logged
  Buffer logging: level debugging, 16695 messages logged

Log Buffer (16384 bytes):

RP/0/RP0/CPU0:Aug 29 06:49:10.642 : exec[65714]: %SECURITY-login-4-AUTHEN_FAILED
  : Failed authentication attempt by user 'user_A' from '161.44.192.231'
RP/0/RP0/CPU0:Aug 29 08:45:58.249 : config[65771]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_A on vty0 (161.44.212.109)
RP/0/RP0/CPU0:Aug 29 08:57:51.183 : exec[65755]: %MGBL-exec-3-LOGIN_AUTHEN : Log
  in Authentication failed. Exiting...
LC/0/1/CPU0:Aug 30 20:17:34.692 : obflmgr[66]: %OS-OBFLMGR-6-COALESCE_START : Co
  alescing started for feature:temperature on device:nvram:(nodeid:0x11). Continuo
  us file size:65522 limit:65536. Historical file size:10864 limit:32768.
LC/0/6/CPU0:Aug 30 21:04:18.913 : obflmgr[66]: %OS-OBFLMGR-6-COALESCE_START : Co
  alescing started for feature:temperature on device:nvram:(nodeid:0x61). Continuo
  us file size:65512 limit:65536. Historical file size:12976 limit:32768.
RP/0/RP0/CPU0:Aug 31 06:47:56.740 : config[65716]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_B on vty0 (10.86.240.72)
RP/0/RP0/CPU0:Aug 31 07:44:12.233 : config[65716]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_B on vty0 (10.86.240.72)
RP/0/RP0/CPU0:Aug 31 07:45:31.728 : config[65716]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_B on vty0 (10.86.240.72)
RP/0/RP1/CPU0:Aug 31 10:13:09.415 : obflmgr[68]: %OS-OBFLMGR-6-COALESCE_START :
  Coalescing started for feature:temperature on device:bootflash:(nodeid:0x211). C
  ontinuous file size:65513 limit:65536. Historical file size:6444 limit:32768.
RP/0/RP0/CPU0:Aug 31 11:46:49.356 : config[65771]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_B on vty1 (10.86.240.72)
RP/0/RP0/CPU0:Sep  1 03:10:13.537 : obflmgr[68]: %OS-OBFLMGR-6-COALESCE_START :
  Coalescing started for feature:temperature on device:bootflash:(nodeid:0x201). C
  ontinuous file size:65520 limit:65536. Historical file size:4164 limit:32768.
RP/0/RP0/CPU0:Sep  1 07:39:13.096 : config[65755]: %MGBL-SYS-5-CONFIG_I : Config
  ured from console by user_B on vty1 (161.44.192.236)
SP/0/SM1/SP:Sep  1 11:48:46.765 : obflmgr[66]: %OS-OBFLMGR-6-COALESCE_START : Co
  alescing started for feature:temperature on device:configflash:(nodeid:0x810). C
  ontinuous file size:65512 limit:65536. Historical file size:0 limit:32768.
SP/0/SM1/SP:Sep  5 02:53:03.819 : obflmgr[66]: %OS-OBFLMGR-6-COALESCE_START : Co
  alescing started for feature:temperature on device:configflash:(nodeid:0x810). C
  ontinuous file size:65512 limit:65536. Historical file size:1392 limit:32768.
SP/0/SM0/SP:Sep  5 02:53:18.837 : obflmgr[66]: %OS-OBFLMGR-6-COALESCE_START : Co
  alescing started for feature:temperature on device:configflash:(nodeid:0x800). C
  ontinuous file size:65512 limit:65536. Historical file size:1716 limit:32768.
```

The example output shows the level of logging. For example, the level of buffer logging is debugging. Ensure that the appropriate logging levels are available for each type of logging (console, monitor, trap, and buffer).

The output from the **show environment** command displays environmental monitor parameters for the system. Verify that the environment parameters are as expected. Environment parameter anomalies are logged in the syslog, so if an environment parameter displayed in the **show environment** command output is not as expected, check the syslog using the **show logging** command. The syslog will provide details on any logged problems.

```
RP/0/RP0/CPU0:router(admin)# show environment

Temperature Information
-----
```

R/S/I	Modules	Inlet Temperature (deg C)	Exhaust Temperature (deg C)	Hotspot Temperature (deg C)
0/1/*	host	29, 28	25, 25	32
	cpu			31
	fabricq0			26
	fabricq1			29
	ingressq			33
	egressq		30	25
	ingresspse			32
	egresspse			27
	jacket	23	23	23
	spa0	18		25, 30
	spa5	23		23

.
.
.

0/RP0/*	host	22	23	22, 31, 25, 23, 25
0/RP1/*	host	22	22	23, 30, 25, 23, 25

.
.
.

0/SM3/*	host	39, 39		36, 44
---------	------	--------	--	--------

Threshold Information

R/S/I	Modules	Sensor	Minor (Lo Hi)	Major (Lo Hi)	Critical (Lo Hi)
0/1/*	host	Inlet0	0,57	-10,67	-15,76
	host	Inlet1	0,64	-10,74	-15,83
	host	Exhaust0	0,57	-10,67	-15,76
	host	Exhaust1	0,59	-10,69	-15,78
	host	Hotspot0	0,63	-10,73	-15,82
	cpu	Hotspot0	0,62	-10,81	-15,91
	fabricq	Hotspot0	0,59	-10,70	-15,79
	fabricq	Hotspot0	0,62	-10,73	-15,82
	ingress	Hotspot0	0,69	-10,79	-15,88
	egressq	Exhaust0	0,64	-10,76	-15,85
	egressq	Hotspot0	0,59	-10,73	-15,82
	ingress	Hotspot0	0,69	-10,80	-15,89
	egressp	Hotspot0	0,60	-10,74	-15,83
	jacket	Inlet0	0,52	-10,60	-15,70
	jacket	Exhaust0	0,60	-10,70	-15,80
	jacket	Hotspot0	0,65	-10,75	-15,85
	spa0	Inlet0	0,50	-10,60	-15,70
	spa0	Hotspot0	0,64	-10,74	-15,84
	spa0	Hotspot1	0,69	-10,79	-15,88
	spa5	Inlet0	0,58	-10,71	-15,81
	spa5	Hotspot0	0,60	-10,73	-15,83

.
.
.

0/RP0/*

```

host      Inlet0      0,46      -10,55      -15,72
host      Hotspot0    0,46      -10,57      -15,74
host      Hotspot1    0,55      -10,67      -15,84
host      Hotspot2    0,50      -10,63      -15,79
host      Hotspot3    0,49      -10,57      -15,74
host      Hotspot4    0,52      -10,66      -15,80
host      Exhaust0    0,47      -10,61      -15,77
.
.
.
0/SM1/*
host      Inlet0      0,68      -10,74      -15,82
host      Inlet1      0,66      -10,72      -15,81
host      Hotspot0    0,59      -10,67      -15,77
host      Hotspot1    0,66      -10,72      -15,81
0/SM1/*
host      3.3V        2970,3630  2805,3795  2640,3960
host      1.8V        1620,1980  1530,2070  1440,2160
host      1.8V        1620,1980  1530,2070  1440,2160
host      2.5V        2250,2750  2125,2875  2000,3000
host      2.5V        2250,2750  2125,2875  2000,3000
.
.
.
Voltage Information
-----
R/S/I  Modules  1.2V    1.25V    1.5V    1.6V    1.8V    2.5V    3.3V    5V
          (mv)    (mv)    (mv)    (mv)    (mv)    (mv)    (mv)    (mv)

0/1/*
host      1254      1240      1494      1790      2548      3250      5000
          1240      1240      1790      2548      3336      5018
          5018
cpu
fabq0     1261      1238      1410      1818      2535      3336      4966
          1240      2538      5000
          5018
fabq1     1261      1238      2538      5000
          1240      4992
ingq      1248      1238      2552      5000
          1240      4992
egrq      1248      2538      5000
i-pse     1261      2538      5000
e-pse     1261      2538      5000
jacket    1530      2522      3316
spa0      1510      2510      3307
spa5      1522      2535      3307
          1526      3308
.
.
.
0/RP0/*
host      1254      1240      1508      1297      1794      2545      3302      5000
          1254      1226      1820      2535      3319      4992
          3285      4992

0/RP1/*
host      1254      1240      1508      1297      1807      2545      3328      5000
          1254      1226      1807      2535      3336      4992
          3268      4940

0/SM0/*
host      1762      2496      3302
          1804      2496

0/SM1/*
host      1762      2483      3285

```

```

                                1804      2496
0/SM2/*
    host                        1762      2496      3285
                                1804      2496
0/SM3/*
    host                        1776      2496      3285
                                1790      2470

```

LED Information

```

-----
0/1/*: Module (host) LED status says: OK
0/1/*: Module (jacket) LED status says: OK
0/1/*: Module (spa0) LED status says: OK
0/1/*: Module (spa5) LED status says: OK
0/6/*: Module (host) LED status says: OK
0/6/*: Module (jacket) LED status says: OK
0/6/*: Module (spa0) LED status says: OK
0/6/*: Module (spa4) LED status says: OK
0/6/*: Module (spa5) LED status says: OK
0/RP0/*: Module (host) LED status says: OK
0/RP0/*: Alarm LED status says: NONE
Rack 0: Upper Fan Tray: LED status : OK
Rack 0: Lower Fan Tray: LED status : OK
0/RP1/*: Module (host) LED status says: OK
0/RP1/*: Alarm LED status says: NONE
0/SM0/*: Module (host) LED status says: OK
0/SM1/*: Module (host) LED status says: OK
0/SM2/*: Module (host) LED status says: OK
0/SM3/*: Module (host) LED status says: OK

```

Fan Information

```

-----
Fan speed (rpm):
      FAN1      FAN2      FAN3      FAN4
Rack 0:
Upper  4842      4882      4842      4882
Lower  4882      4842      4923      4842

```

Power Supply Information

```

-----
Power-Supply      Voltage      Current
                   (V)              (A)
Zone 1:           [A], [B]      54.867, 54.377  4.596,  4.387
Zone 2:           [A], [B]      54.573, 53.985  6.894,  5.745
Zone 3:           [A], [B]      55.161, 54.279  4.387,  4.491

Total Current:    30.500 A
Total Power   : 1677.500 W

```

The example output shows the system environmental monitor parameter variables.

The output from the **show context** command displays core dump context information. See the [“show context Command”](#) section on page 1-11 for more information on the **show context** command output.

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

```
node:      node0_1_CPU0
-----
```

```
Crashed pid = 61524 (pkg/bin/tcam_mgr)
```

```
Crashed tid = 1
Crash time: Wed Apr 05, 2006: 18:27:26
Core for process at harddisk:/dumper/first.tcam_mgr.abort.node0_1_CPU0.ppc.Z
```

Stack Trace

```
#0 0xfc1d3fa0
#1 0xfc1c6340
#2 0xfc1c5364
#3 0xfc1c542c
#4 0x48210930
#5 0x482110b8
#6 0x48212ba4
#7 0x48203dd8
#8 0x4820c61c
#9 0xfc1557ec
#10 0xfc15573c
#11 0xfc152fb8
#12 0x4820d140
```

Registers info

	r0	r1	r2	r3
R0	00000000	481ff7b0	4824a55c	00000000
	r4	r5	r6	r7
R4	0000f054	00000001	00000006	00000000
	r8	r9	r10	r11
R8	00000000	fc220000	481fffc0	00000000
	r12	r13	r14	r15
R12	4823be90	4824a4a0	48230000	00000000
	r16	r17	r18	r19
R16	00000048	00000001	00000019	48256520
	r20	r21	r22	r23
R20	00000000	00000000	00000003	00000045
	r24	r25	r26	r27
R24	00000003	00000000	00000003	4825dc34
	r28	r29	r30	r31
R28	00000006	0000f054	48254064	481ff810
	cnt	lr	msr	pc
R32	00000000	fc1c6340	0000d932	fc1d3fa0
	cnd	xer		
R36	28004024	00000008		

DLL Info

DLL path	Text addr.	Text size	Data addr.	Data size	Version
/hfr-os-3.3.90/lib/libinfra.dll	0xfc142000	0x00034200	0xfc1343b8	0x00000bbc	0
/lib/libc.dll	0xfcla8000	0x00079dd8	0xfc222000	0x00002000	0

Crash Package Infomation

```
Package: hfr-mgbl, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-mcast, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-mpls, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-rout, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-k9sec, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-lc, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-fwdg, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-admin, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
Package: hfr-base, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/workspace for c2.95.3-p8
```



```
kospace for c2.95.3-p8
Package: hfr-os-mbi, Source: By edde-bld1 in /vws/aga/production/3.3.90.1I/hfr/w
orkspace for c2.95.3-p8
```

```
node:      node0_6_CPU0
-----
```

```
node:      node0_RP0_CPU0
-----
```

```
node:      node0_RP1_CPU0
-----
```

The example output shows that the pkg/bin/tcam_mgr process crashed.

The output from the **show memory** command displays information about the memory available on the router after the system image decompresses and loads. Verify that the expected memory is available or installed. Ensure that all memory regions have adequate free space available.

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

```
Physical Memory: 4.000G total
Application Memory : 3.857G (3.455G available)
Image: 17.880M (bootram: 17.880M)
Reserved: 128.000M, IOMem: 1.980G, flashfsys: 0
Shared window infra_ital: 323.628K
Shared window ipv4_fib: 1.003M
Shared window ifc-mpls: 961.714K
Shared window ifc-ipv6: 1.189M
Shared window ifc-ipv4: 1.251M
Shared window ifc-protomax: 641.714K
Shared window infra_statsd: 3.714K
Shared window aib: 203.687K
Shared window PFI_IFH: 155.652K
Shared window squid: 2.152M
Shared window atc_cache: 35.671K
Total shared window: 7.867M
Allocated Memory: 170.406M
Program Text: 21.242M
Program Data: 1.761M
Program Stack: 6.878M
```

The example output shows that there is 3.455 gigabits of application memory available.

The output from the **show running-config** command displays the current running configuration. Verify that the contents of the current running configuration are what is expected.

```
RP/0/RP0/CPU0:router# show running-config
```

```
Building configuration...
!! Last configuration change at 18:56:31 UTC Tue Feb 28 2006 by user_A
!
hostname CRS-8_X1
line console
  exec-timeout 120 0
  session-timeout 120
!

line default
  exec-timeout 120 0
  session-timeout 120
!
telnet vrf default ipv4 server max-servers no-limit
domain ipv4 host x1 172.16.52.72
```

```

domain ipv4 host x2 172.16.52.77
domain ipv4 host xe1 172.16.52.73
domain ipv4 host xe2 172.16.52.78
domain-lookup
vty-pool default 0 25
ipv4 virtual address 172.16.52.72 255.255.255.0
interface Loopback0
  ipv4 address 10.10.20.0 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
  description Connected to RTR RTR
  ipv4 address 172.16.52.70 255.255.255.0
!
interface MgmtEth0/RP1/CPU0/0
  description Connected to Lab LAN
  ipv4 address 172.16.52.71 255.255.255.0
!
interface GigabitEthernet0/1/5/0
  description Connected to CRS-8_X2 GE 0/1/5/0
  ipv4 address 10.50.40.0 255.255.255.0
!
interface GigabitEthernet0/1/5/1
  description Connected to C12810_XF GE 5/2
  ipv4 address 10.50.56.0 255.255.255.0
  negotiation auto
!
interface GigabitEthernet0/1/5/2
  shutdown
!
interface GigabitEthernet0/1/5/3
  shutdown
!
interface GigabitEthernet0/1/5/4
  shutdown
!
interface GigabitEthernet0/1/5/5
  shutdown
!
interface GigabitEthernet0/1/5/6
  shutdown
!
interface GigabitEthernet0/1/5/7
  shutdown
!
interface GigabitEthernet0/6/5/0
  description Connected to C7304_XR1 GE2
  ipv4 address 10.55.12.0 255.255.255.0
!
interface GigabitEthernet0/6/5/1
  description Connected to CRS-8_X2 GE 0/6/5/1
  ipv4 address 10.55.48.0 255.255.255.0
!
.
.
.
!
controller SONET0/1/0/0
  delay trigger line 100
  clock source internal
!
controller SONET0/1/0/1
  clock source internal
!
controller SONET0/6/0/0

```

```

delay trigger line 100
clock source internal
!
controller SONET0/6/0/1
clock source internal
!
controller SONET0/6/4/4
clock source internal
!
controller SONET0/6/4/5
clock source internal
!
controller SONET0/6/4/6
clock source internal
!
controller SONET0/6/4/7
clock source internal
!
router static
address-family ipv4 unicast
0.0.0.0/0 172.16.52.1
!
!
ssh server
end

```

The example output displays the contents of the current running configuration.

The **show system verify start** command starts the system verification process and the **show system verify report** generates the output from the system verification process. The output allows you to verify that the system parameters are as expected.

```
RP/0/RP0/CPU0:router# show system verify start
```

```

Storing initial router status ...
done.
RP/0/RP0/CPU0:router#

```

```
RP/0/RP0/CPU0:router# show system verify report
```

```

Getting current router status ...
System Verification Report
=====
- Verifying Memory Usage
- Verified Memory Usage                : [OK]
- Verifying CPU Usage
- Verified CPU Usage                    : [OK]

- Verifying Blocked Processes
- Verified Blocked Processes            : [OK]
- Verifying Aborted Processes
- Verified Aborted Processes            : [OK]
- Verifying Crashed Processes
- Verified Crashed Processes            : [OK]

- Verifying LC Status
- Verified LC Status                    : [OK]
- Verifying QNET Status
Unable to get current LC status info
- Verified QNET Status                  : [FAIL]

- Verifying GSP Fabric Status
- Verified GSP Fabric Status            : [OK]
- Verifying GSP Ethernet Status

```

```

- Verified GSP Ethernet Status : [OK]

- Verifying POS interface Status
- Verified POS interface Status : [OK]
- Verifying TenGigE interface Status
- Verified TenGigE interface Status : [OK]

- Verifying TCP statistics
- Verified TCP statistics : [OK]
- Verifying UDP statistics
  tcp_udp_raw WARNING messages for router
  UDP Packets sent has not increased during this period.
- Verified UDP statistics : [WARNING]
- Verifying RAW statistics
- Verified RAW statistics : [OK]

- Verifying RIB Status
- Verified RIB Status : [OK]
- Verifying CEF Status
- Verified CEF Status : [OK]
- Verifying CEF Consistency Status
- Verified CEF Consistency Status : [OK]
- Verifying BGP Status
- Verified BGP Status : [OK]
- Verifying ISIS Status
- Verified ISIS Status : [OK]
- Verifying OSPF Status
- Verified OSPF Status : [OK]

- Verifying Syslog Messages
- Verified Syslog Messages : [OK]

```

System may not be stable. Please look into WARNING messages.

The example output compares the system from the time the **show system verify start** command took the first snapshot to the snapshot taken of the system when the **show system verify report** command took the second snapshot and generated the comparison. If there are no changes, [OK] is displayed. If there are changes between the first and second snapshot, the specific change is noted and marked with [WARNING] or [FAIL].

The **show interface brief** command displays the usability status of the configured interfaces. Verify that all expected interfaces are listed. For an interface to be usable, both the interface hardware (Status) and line protocol must be up. The protocol is Up if the interface can provide two-way communication.

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

Interface	IP-Address	Status	Protocol
Loopback0	10.10.20.1	Up	Up
MgmtEth0/RP0/CPU0/0	172.29.52.70	Up	Up
POS0/1/0/0	10.50.4.1	Up	Up
POS0/1/0/1	10.50.32.1	Up	Up
POS0/1/0/2	unassigned	Shutdown	Down
POS0/1/0/3	unassigned	Shutdown	Down
GigabitEthernet0/1/5/0	10.50.40.1	Up	Up
GigabitEthernet0/1/5/1	10.50.56.1	Up	Up
GigabitEthernet0/1/5/2	unassigned	Shutdown	Down
GigabitEthernet0/1/5/3	unassigned	Shutdown	Down
GigabitEthernet0/1/5/4	unassigned	Shutdown	Down
GigabitEthernet0/1/5/5	unassigned	Shutdown	Down
GigabitEthernet0/1/5/6	unassigned	Shutdown	Down
GigabitEthernet0/1/5/7	unassigned	Shutdown	Down
POS0/6/0/0	10.50.8.1	Up	Up
POS0/6/0/1	10.50.36.1	Up	Up

POS0/6/0/2	unassigned	Shutdown	Down
POS0/6/0/3	unassigned	Shutdown	Down
POS0/6/4/0	unassigned	Shutdown	Down
POS0/6/4/1	unassigned	Shutdown	Down
POS0/6/4/2	unassigned	Shutdown	Down
POS0/6/4/3	unassigned	Shutdown	Down
POS0/6/4/4	10.50.52.1	Up	Up
POS0/6/4/5	10.50.28.1	Up	Up
POS0/6/4/6	10.50.104.1	Up	Up
POS0/6/4/7	10.50.44.1	Up	Up
GigabitEthernet0/6/5/0	10.50.12.1	Up	Up
GigabitEthernet0/6/5/1	10.50.48.1	Up	Up
GigabitEthernet0/6/5/2	unassigned	Shutdown	Down
GigabitEthernet0/6/5/3	unassigned	Shutdown	Down
GigabitEthernet0/6/5/4	unassigned	Shutdown	Down
GigabitEthernet0/6/5/5	unassigned	Shutdown	Down
GigabitEthernet0/6/5/6	unassigned	Shutdown	Down
GigabitEthernet0/6/5/7	unassigned	Shutdown	Down
MgmtEth0/RP1/CPU0/0	172.29.52.71	Up	Up

The example output displays IP addresses, status, and protocol status for each interface. The output shows that all assigned interfaces (interfaces that are configured with IP addresses) have a interface hardware status and line protocol status of up.

Validating and Troubleshooting Cisco IOS XR Software Package Installation

The Cisco IOS XR software is divided into software packages allowing you to select which features run on your router. Each package contains the components to perform a specific set of router functions, such as routing, security, or Modular Services card support. Bundles are groups of packages that can be downloaded as a set. For example, the Unicast Routing Core Bundle provides six packages for use on every router.

This section provides information on how to validate and troubleshoot the Cisco IOS XR software package installation. The following sections are provided:

- [Verifying the Software Version, page 1-27](#)
- [Validating the Installation, page 1-30](#)

Verifying the Software Version

To verify the Cisco IOS XR software version, perform the following procedure.

SUMMARY STEPS

1. **show version**
2. **show install**

DETAILED STEPS

	Command or Action	Purpose
Step 1	show version Example: RP/0/RP0/CPU0:router# show version	Displays a variety of system information, including hardware and software version, router uptime, boot settings (configuration register), and active software. Determine if all expected packages are installed and the current software versions are the expected versions. If the expected packages are not installed or are not the expected version, install the correct package. See <i>Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router</i> for information on installing and upgrading Cisco IOS XR software packages.
Step 2	show install Example: RP/0/RP0/CPU0:router# show install	Displays a list of all installed and active packages on each node. Determine if the expected packages are installed on each node. If the software or active package versions are not as expected for a node, the package is not compatible with the node for which it is being activated, or the package being activated is not compatible with the current active software set, install the correct software or package on the node. See <i>Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router</i> for information on installing and upgrading Cisco IOS XR software packages.

The following example shows that the Cisco IOS XR software and active packages are version 3.3.0.

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

Cisco IOS XR Software, Version 3.3.0
Copyright (c) 2006 by cisco Systems, Inc.

ROM: System Bootstrap, Version 1.32(20050525:193559) [CRS-1 ROMMON],

CRS-8_X1 uptime is 2 weeks, 4 days, 23 hours, 27 minutes
System image file is "disk0:hfr-os-mbi-3.3.0/mbihfr-rp.vm"

cisco CRS-8/S (7457) processor with 4194304K bytes of memory.
7457 processor at 1197Mhz, Revision 1.2

16 Packet over SONET/SDH network interface(s)
16 SONET/SDH Port controller(s)
2 Ethernet/IEEE 802.3 interface(s)
16 GigabitEthernet/IEEE 802.3 interface(s)
2043k bytes of non-volatile configuration memory.
38079M bytes of hard disk.
1000592k bytes of ATA PCMCIA card at disk 0 (Sector size 512 bytes).
1000640k bytes of ATA PCMCIA card at disk 1 (Sector size 512 bytes).

Package active on node 0/1/SP:
hfr-diags, V 3.3.0, Cisco Systems, at disk0:hfr-diags-3.3.0
  Built on Tue Jan 31 13:32:40 UTC 2006
  By edde-bld1 in /files/3.3.0/hfr/workspace for c2.95.3-p8
```

```
hfr-admin, V 3.3.0, Cisco Systems, at disk0:hfr-admin-3.3.0
  Built on Tue Jan 31 10:02:02 UTC 2006
  By edde-bld1 in /files/3.3.84.2I/hfr/workspace for c2.95.3-p8

hfr-base, V 3.3.0, Cisco Systems, at disk0:hfr-base-3.3.0
  Built on Tue Jan 31 09:48:20 UTC 2006
  By edde-bld1 in /files/3.3.84.2I/hfr/workspace for c2.95.3-p8
.
.
.
```

The following example shows that the Cisco IOS XR software and active packages are version 3.3.0.

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

```
Node 0/1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/1/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/6/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/6/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/RP0/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Active Packages:
    disk0:hfr-mgbl-3.3.0
    disk0:hfr-k9sec-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/RP1/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Active Packages:
    disk0:hfr-mgbl-3.3.0
    disk0:hfr-k9sec-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/SM0/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM2/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0
```

```

Node 0/SM3/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:comp-hfr-mini-3.3.0

```

The example output shows that all the active Cisco IOS XR software packages are version 3.3.0. If there is an expected package missing or an active package is not an expected package, install and activate the missing package or upgrade the unexpected package to the appropriate package. See the *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for details on installing, activating, and upgrading software packages.

Validating the Installation

Validate the Cisco IOS XR software package installation to ensure the packages were installed correctly. The following commands are used to validate the currently installed software packages:

- [install verify Command, page 1-30](#)
- [show install active Command, page 1-32](#)
- [show install committed Command, page 1-33](#)

install verify Command

Use the **install verify** command to verify the consistency of a previously installed software set with the package file from which it originated.

This command can be used as a debugging tool to verify the validity of the files that constitute the packages to determine if there are any corrupted files. The command is also used to check that the install infrastructure is up and running and to determine if all files are expected. If there are corrupted files, see *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on deactivating and removing software packages and adding and activating software packages.



Note

The **install verify** command can take up to two minutes per package to process.



Note

The **install verify** command ignores secure domain router (SDR) boundaries and performs the operation in global scope.

The following example shows the output of the **install verify** command. The output is used to verify the consistency of a previously installed software set with the package file from which it originated.

```

RP/0/RP0/CPU0:router(admin)# install verify

Install operation 6 'install verify' started by user 'user_a' at 07:25:16 UTC
Tue Mar 07 2006.
The install operation will continue asynchronously.
RP/0/RP1/CPU0:router(admin)#Info:      This operation can take up to 2 minutes.
Info:      Please be patient.
Info:      Verify operation successful, no anomalies found.
Info:      Node 0/1/SP
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.

```



```

Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/1/CPU0
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-mcast-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-mpis-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-lc-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-fwdg-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/6/SP
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/6/CPU0
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-mcast-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-mpis-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-lc-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-fwdg-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/RP0/CPU0
Info:      [SUCCESS] /disk0/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-k9sec-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mgbl-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-rout-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mcast-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mpis-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-lc-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-fwdg-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/RP1/CPU0
Info:      [SUCCESS] /disk0/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-k9sec-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mgbl-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-rout-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mcast-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-mpis-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-lc-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-fwdg-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /disk0/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/SM0/SP
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/SM1/SP
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/SM2/SP
Info:      [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:      [SUCCESS] /bootflash/hfr-os-mpi-3.3.0: Verification Successful.
Info:      Node 0/SM3/SP

```

```

Info:          [SUCCESS] /bootflash/hfr-diags-3.3.0: Verification Successful.
Info:          [SUCCESS] /bootflash/hfr-admin-3.3.0: Verification Successful.
Info:          [SUCCESS] /bootflash/hfr-base-3.3.0: Verification Successful.
Info:          [SUCCESS] /bootflash/hfr-os-mbi-3.3.0: Verification Successful.
Install operation 6 completed successfully at 07:30:48 UTC Tue Mar 07 2006.

```

show install active Command

Use the **show install active** command to display active software packages. Verify that the command output matches the output of the **show install committed** command. If the output does not match, when you reload the router, the software displayed in the **show install committed** command output is the software that will be loaded. For example, the following output shows two different software package versions, one is the active version and the other is the committed version, so when the router reloads, The 3.2.6 version will be loaded even though 3.3.0 is the currently active version on 0/RP0/CPU0:

```
RP/0/RP0/CPU0:router(admin)# show install active location 0/rp0/cpu0
```

```

Node 0/RP0/CPU0 [RP] [SDR: Owner]
Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
Active Packages:
  disk0:hfr-infra-test-3.3.0
  disk0:hfr-mgbl-3.3.0
  disk0:hfr-mcast-3.3.0
  disk0:hfr-mpls-3.3.0
  disk0:hfr-k9sec-3.3.0
  disk0:comp-hfr-mini-3.3.0

```

```
RP/0/RP0/CPU0:router(admin)# show install committed location 0/rp0/cpu0
```

```

Node 0/RP0/CPU0 [RP] [SDR: Owner]
Boot Image: /disk0/hfr-os-mbi-3.2.6/mbihfr-rp.vm
Committed Packages:
  disk0:hfr-mgbl-3.2.6
  disk0:hfr-mcast-3.2.6
  disk0:hfr-mpls-3.2.6
  disk0:hfr-k9sec-3.2.6
  disk0:comp-hfr-mini-3.2.6

```

If the expected active software packages are not displayed, install the packages (if required) and activate the packages. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on installing and activating Cisco IOS XR software packages. The following example output shows the active packages for all cards in a router:

```
RP/0/RP0/CPU0:router# show install active
```

```

Node 0/1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/1/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/6/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0

```

```

disk0:comp-hfr-mini-3.3.0

Node 0/6/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/RP0/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:hfr-mgbl-3.3.0
    disk0:hfr-k9sec-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/RP1/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:hfr-mgbl-3.3.0
    disk0:hfr-k9sec-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/SM0/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/SM1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/SM2/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

Node 0/SM3/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Active Packages:
    disk0:hfr-diags-3.3.0
    disk0:comp-hfr-mini-3.3.0

```

The active packages for each node are on disk0, and for all nodes, the composite package hfr-os-mbi-3.3.0 is active. Additional packages shown are optional packages that have been activated after the initial loading of the Cisco IOS XR Unicast Routing Core Bundle.

show install committed Command

Use the **show install committed** command to display committed software packages. The committed software packages are the software packages that will be booted on a router reload.

Committed packages are the packages that are persistent across router reloads. If you install and activate a package, it remains active until the next router reload. If you commit a package set, all packages in that set remain active across router reloads until the package set is replaced with another committed package

set. The **show install committed** command is useful to ensure software is installed and committed after a router reload. If the expected software is not installed and committed, see *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on installing and committing Cisco IOS XR software packages.

The following command output shows the committed software packages on all card in the router.

```
RP/0/RP0/CPU0:router# show install committed

Secure Domain Router: Owner

Node 0/1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/1/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/6/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/6/CPU0 [LC] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/lc/mbihfr-lc.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/RP0/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/RP1/CPU0 [RP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/mbihfr-rp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM0/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM1/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM2/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0

Node 0/SM3/SP [SP] [SDR: Owner]
  Boot Image: /disk0/hfr-os-mbi-3.3.0/sp/mbihfr-sp.vm
  Committed Packages:
    disk0:comp-hfr-mini-3.3.0
```

The active packages for each node are on disk0, and for all nodes, the composite package hfr-os-mbi-3.3.0 is active.

Validating and Troubleshooting Cisco IOS XR Software Configuration

Validating the Cisco IOS XR software configuration includes collecting configuration information on the router to determine configuration changes and verifying the current running configuration. When a configuration fails during a commit, the failed configuration can be viewed to help determine why the configuration was not committed.

The following sections are provided:

- [Local and Global Configurations, page 1-35](#)
- [Collecting Configuration Information, page 1-37](#)
- [Verifying the Running Configuration, page 1-38](#)
- [Using the show configuration failed Command, page 1-47](#)

Local and Global Configurations

Configuration data is split between global (shared) and local configurations. Configurations are held locally to the appropriate node. For example, the system configuration is distributed to the node it belongs on. The routing protocol configurations that are shared for all nodes are part of the SysDB shared plane running on the dLRSC RP node.

The local plane configurations, such as interface-specific configuration, belong to the local plane SysDB running on each node. Every node has a data store containing the local data for that node (local plane), including configuration and operational data for the local interfaces. There is also a data store containing the shared data (shared plane) which is primarily used by RP and DRP applications, but accessible to all nodes.

Each SysDB item is categorized as either 'local' or 'shared'. Local data is that primarily of interest to a single node and shared data is everything else. Since almost all line card usage involves only local data, their SysDB clients only use their local server which minimizes remote inter-process communication (IPC).

When troubleshooting configurations, you need to determine whether the problem is local or shared (global). To view the local configuration, use the **show running-config interface *** command.

```
RP/0/RP0/CPU0:router# show running-config interface *
```

```
interface Bundle-Ether28
 bundle minimum-active bandwidth 1000000
 bundle minimum-active links 1
 ipv4 address 10.12.28.1 255.255.255.0
 description Connected to P2_CRS-8 Bundle-Ether 28
!
interface Bundle-Ether28.1
 dot1q vlan 29
 ipv4 address 10.12.29.1 255.255.255.0
 description Connected to P2_CRS-8 Bundle-Ether 28.1
.
.
.
interface Bundle-POS24
 bundle minimum-active bandwidth 2488320
 bundle minimum-active links 1
 ipv4 address 10.12.24.1 255.255.255.0
 description Connected to P2_CRS-8 Bundle-POS 24
```

```

!
interface Loopback0
  ipv4 address 10.1.1.1 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
  description Connected to Lab LAN
  ipv4 address 172.29.52.70 255.255.255.0
!
interface MgmtEth0/RP1/CPU0/0
  description Connected to Lab LAN
  ipv4 address 172.29.52.71 255.255.255.0
!
interface GigabitEthernet0/1/5/0
  description Connected to P2_CRS-8 GE 0/1/5/0
  ipv4 address 10.12.16.1 255.255.255.0
!
interface GigabitEthernet0/1/5/1
  description Connected to P4_C12810 GE 5/2
  ipv4 address 10.14.8.1 255.255.255.0
!
interface GigabitEthernet0/1/5/2
  description Connected to PE6_C12406 GE 0/4/0/1
  ipv4 address 10.16.4.1 255.255.255.0
!
interface GigabitEthernet0/1/5/3
  shutdown
.
.
.
interface POS0/1/4/1
  description Connected to P2_CRS-8 POS 0/1/4/1
  bundle id 24 mode active
!
interface POS0/1/4/2
  description Connected to P2_CRS-8 POS 0/1/4/2
  ipv4 address 10.12.32.1 255.255.255.0
!
interface POS0/1/4/3
  description Connected to P2_CRS-8 POS 0/1/4/3
  ipv4 address 10.12.36.1 255.255.255.0
.
.
.
controller SONET0/6/4/6
  clock source internal
!

```

The output displays all the configured interfaces on the node.

Use the **show sysdb trace** commands to display the contents of the SysDB after a configuration change. The trace information includes a history of any changes to the running configuration. You can specify either a local node or the shared plane.

The following example output shows the contents of the SysDB local plane:

```
RP/0/RP0/CPU0:router# show sysdb trace verification location 0/5/cpu0 reverse
```

Timestamp	path	jid	tid	reg handle	connid	action
656 wrapping entries (4096 possible, 3440 filtered, 6460 total)						
Aug 29 06:14:38.443		116	1	20	38446	apply reply
'--'						
Aug 29 06:14:38.442		116	1	20	1139	Apply/abort called
'cfg/if/act/POS0_5_0_0/keepalive'						

```

Aug 29 06:14:38.441      116      1      20      1139      verify reply: accept
'__'
Aug 29 06:14:38.438      116      1      20      1139      Verify called
'cfg/if/act/POS0_5_0_0/keepalive'

```

The following example output shows the contents of the SysDB shared plane:

```
RP/0/RP0/CPU0:router# show sysdb trace verification shared-plane reverse
```

```

Timestamp      jid      tid  reg handle  connid      action
      path
4 wrapping entries (4096 possible, 4092 filtered, 904284 total)
Aug 29 06:16:53.244      526      1      880      12043      apply reply
'__'
Aug 29 06:16:53.229      526      1      880      1111      Apply/abort called
'cfg/gl/aaa/tacacs/source-interface'
Aug 29 06:16:53.225      526      1      880      1111      verify reply: accept
'__'
Aug 29 06:16:53.214      526      1      880      1111      Verify called
'cfg/gl/aaa/tacacs/source-interface'

```

The **show processes location *node-id* | include sysdb** command displays all active SysDB processes for a specified node. See [Chapter 1, “General Troubleshooting,”](#) for information on troubleshooting processes.

Collecting Configuration Information

Collecting configuration information allows you to determine if changes to the system have occurred. It also allows you to determine if these changes could impact the system. The following commands allow you to determine if there was an unknown commit, if there was a commit that overwrote a previous configuration, or there are configuration changes that should be removed from the running configuration.

- **show config commit history**—the command output displays information about the last (up to) 1000 commits, of which only the last (up to) 100 commits are available for rollback operations.
- **show configuration commit changes {[since] *commit-id* | last *number-of-commits*} [diff]**—the command output displays changes made to the running configuration by previous configuration commits.

```
RP/0/RP0/CPU0:router# show configuration commit changes since 1000000319
```

```

Wed May 17 09:30:27.877 UTC
Building configuration...
no logging console
no domain ipv4 host ce1
no domain ipv4 host ce2
domain ipv4 host ce6 172.29.52.73
domain ipv4 host ce7 172.29.52.78
no domain ipv4 host pe1
no domain ipv4 host pe2
domain ipv4 host pe6 172.29.52.128
domain ipv4 host pe7 172.29.52.182
interface GigabitEthernet0/1/5/1
  no negotiation
!
end

```

- **show configuration commit list** [*number-of-commits*] [**detail**]*—*the command output displays a list of the commit IDs (up to 100) available for rollback.

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

```
Wed May 17 09:31:21.727 UTC
SNo. Label/ID   User      Line      Client      Time Stamp
~~~~ ~~~~~~    ~~~~~    ~~~~~    ~~~~~~
1    1000000324   userA     vty0      CLI         16:50:33 UTC Wed May 10 2006
2    1000000323   userA     vty0      CLI         16:49:51 UTC Wed May 10 2006
3    1000000322   userB     vty0      CLI         16:48:05 UTC Wed May 10 2006
4    1000000321   userC     vty2      CLI         19:11:26 UTC Wed May 03 2006
5    1000000320   userA     vty2      CLI         19:10:45 UTC Wed May 03 2006
6    1000000319   userB     vty2      CLI         18:03:01 UTC Wed May 03 2006
7    1000000318   userB     vty2      CLI         18:02:43 UTC Wed May 03 2006
8    1000000317   userB     vty2      CLI         18:02:38 UTC Wed May 03 2006
9    1000000316   userC     vty2      CLI         17:59:16 UTC Wed May 03 2006
10   1000000315   userC     vty2      CLI         17:46:38 UTC Wed May 03 2006
11   1000000314   userA     vty2      CLI         15:40:04 UTC Wed May 03 2006
12   1000000313   userA     vty2      CLI         13:05:09 UTC Wed May 03 2006
13   1000000312   userD     con0_RP0_C CLI         13:49:31 UTC Mon May 01 2006
```

- **commit confirmed** *minutes**—*the command commits the configuration on a trial basis for a minimum of 30 seconds and a maximum of 300 seconds (5 minutes). During the trial configuration period, enter commit to confirm the configuration. If commit is not entered, then the system will revert to the previous configuration when the trial time period expires.

Verifying the Running Configuration

To verify the running configuration, perform the following procedure.

SUMMARY STEPS

1. **configure**
1. **show running-config**
2. **describe** *hostname* *hostname*
3. **end**
4. **show sysdb trace verification shared-plane** | **include** *path*
5. **show sysdb trace verification location** *node-id*
6. **show cfgmgr trace**
7. **show config commit history**
8. **show config commit changes**
9. **show config failed startup**
10. **cfs check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 1	show running-config Example: RP/0/RP0/CPU0:router(config)# show running-config	Displays the contents of the running configuration. Verify that the running configuration is as expected.
Step 2	describe hostname <i>hostname</i> Example: RP/0/RP0/CPU0:router(config)# describe hostname router_A	Determines the path.
Step 3	end Example: RP/0/RP0/CPU0:router(config)# end	Saves configuration changes. <ul style="list-style-type: none"> When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
Step 4	show sysdb trace verification shared-plane include <i>path</i> Example: RP/0/RP0/CPU0:router# show sysdb trace verification shared-plane include gl/a/hostname	Displays details of recent verification sysDB transactions and changes on the shared plane allowing you to verify whether the configuration was verified correctly. Specifying the path filters the data to display only the sysDB path for the router. Verify that changes to the SysDB were verified and accepted.
Step 5	show sysdb trace verification location <i>node-id</i> Example: RP/0/RP0/CPU0:router# show sysdb trace verification location 0/3/CPU0	Displays details of recent verification sysDB transactions and changes on local plane configurations. Verify that changes to the SysDB were verified and accepted.

	Command or Action	Purpose
Step 6	show cfmgr trace Example: RP/0/RP0/CPU0:router# show cfmgr trace	Displays cfmgr trace information.
Step 7	show configuration history commit Example: RP/0/RP0/CPU0:router# show configuration history commit	Displays a list of historical changes to the configuration. Verify that the timeline of changes is as expected.
Step 8	show configuration commit changes Example: RP/0/RP0/CPU0:router# show configuration commit changes	Displays detailed committed configuration history information. Verify that the history information is as expected.
Step 9	show configuration failed startup Example: RP/0/RP0/CPU0:router# show configuration failed startup	Displays information on any configurations that failed during startup.
Step 10	cfs check Example: RP/0/RP0/CPU0:router# cfs check	Checks the current configuration to see if there are any missing configurations.

Examples

The following example shows the output of the **show running-config** command:

```
RP/0/RP0/CPU0:router# show running-config

Thu May 18 13:13:05.187 UTC
Building configuration...
!! Last configuration change at 12:51:32 UTC Wed May 17 2006 by user_A
!
snmp-server traps fabric plane
hostname router_A
line console
  exec-timeout 600 0
  session-timeout 600
!
line default
  exec-timeout 600 0
  session-timeout 600
!
telnet vrf default ipv4 server max-servers no-limit
domain ipv4 host p1 192.0.2.72
domain ipv4 host p2 192.0.2.77
domain ipv4 host ce6 192.0.2.73
domain ipv4 host ce7 192.0.2.78
domain ipv4 host pe6 192.0.2.128
domain ipv4 host pe7 192.0.2.182
domain-lookup
vty-pool default 0 25
ipv4 virtual address 192.0.2.72 255.255.255.0
```

```

class-map match-any llg
  match mpls experimental topmost 5
  match precedence critical
!
class-map match-any default
  match any
!
class-map match-any business
  match mpls experimental topmost 5
  match precedence flash
!
policy-map fabric-qos
  class llg
    priority
  !
  class business
    bandwidth remaining percent 65
  !
  class default
    bandwidth remaining percent 35
  !
!
interface Loopback0
  ipv4 address 10.1.1.1 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
  description Connected to router LAN
  ipv4 address 192.0.2.70 255.255.255.0
!
.
.
.
!
interface POS0/6/0/3
  shutdown
!
interface POS0/6/4/0
  shutdown
!
interface POS0/6/4/1

```

The output is used to determine if the configuration is as expected.

In the following example, the path to SysDB where the configuration is stored in the database is displayed.

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

The command is defined in shellutil.parser

```
Node 0/RP0/CPU0 has file shellutil.parser for boot package /disk0/hfr-os-mbi-3.3
.O/mbihfr-rp.vm from hfr-base
```

Package:

```

  hfr-base
    hfr-base V3.3.0[2I]  Base Package
    Vendor  : Cisco Systems
    Desc    : Base Package
    Build   : Built on Mon May  1 06:27:09 UTC 2006
    Source  : By edde-bld1 in /vws/3.3.0./file for
c2.95.3-p8
    Card(s): RP, DRP, DRPSC, OC3-POS-4, OC12-POS, GE-3, OC12-POS-4, OC48-POS
, E3-OC48-POS, E3-OC12-POS-4, E3-OC3-POS-16, E3-OC3-POS-8, E3-OC3-POS-4, E3-OC48
-CH, E3-OC12-CH-4, E3-OC3-CH-16, E3-GE-4, E3-OC3-ATM-4, E3-OC12-ATM-4, E5-CEC, L
C, SP

```

```

Restart information:
  Default:
    parallel impacted processes restart

Component:
  shellutil V[r33x/1] Common shell utility applications

File: shellutil.parser

```

User needs ALL of the following taskids:

```
root-lr (READ WRITE)
```

```

It will take the following actions:
  Create/Set the configuration item:
    Path: gl/a/hostname
    Value: pl_CRS-8

```

The output shows that the path is gl/a/hostname.

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

In the following example, the verification details for the specified hostname is displayed.

```
RP/0/RP0/CPU0:router# show sysdb trace verification shared-plane | include gl/a/hostname
```

```

May 18 19:16:17.143      340      3      210      962      Apply/abort called
      'cfg/gl/a/hostname'
May 18 19:16:17.132      340      3      210      962      Verify called
      'cfg/gl/a/hostname'
May 18 19:16:17.126      340      3      210      962      Apply/abort called
      'cfg/gl/a/hostname'
May 18 19:16:17.109      340      3      210      962      Verify called
      'cfg/gl/a/hostname'
May 18 18:43:16.065      340      3      210      962      register
      'cfg/gl/a/hostname'
May 18 18:41:41.048      340      3      16      362      register
      'cfg/gl/a/hostname'

```

The output shows that changes to the SysDB shared plane were verified and accepted.

In the following example, the verification details for the specified location is displayed.

```
RP/0/RP0/CPU0:router# show sysdb trace verification location 0/3/CPU0
```

```

Timestamp      jid      tid  reg handle  connid  action
      path
323 wrapping entries (4096 possible, 299 filtered, 622 total)
Jul  7 20:10:36.212      260      1      90      8782      apply reply
      '---'
Jul  7 20:10:35.476      260      1      90      4912      Apply/abort called
      'cfg/if/act/GigabitEthernet0_3_4_0.1/a/sub_vlan/0x2/_____/Gigab
itEthernet0_3_4_0/_____'
Jul  7 20:10:35.475      260      1      90      4912      verify reply: accep
t      '---'
Jul  7 20:10:35.471      260      1      90      4912      Verify called
      'cfg/if/act/GigabitEthernet0_3_4_0.1/a/sub_vlan/0x2/_____/Gigab
itEthernet0_3_4_0/_____'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '---'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '---'
Jul  7 20:10:35.471      144      1      4      8782      apply reply

```

```

      '___'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '___'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '___'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '___'
Jul  7 20:10:35.471      144      1      4      8782      apply reply
      '___'
Jul  7 20:10:35.470      144      1      4      474      Apply/abort batch e
nded
Jul  7 20:10:35.470      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_0/ord_x/im/shutdown'
Jul  7 20:10:35.470      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_1/ord_x/im/shutdown'
Jul  7 20:10:35.470      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_2/ord_x/im/shutdown'
Jul  7 20:10:35.470      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_3/ord_x/im/shutdown'
Jul  7 20:10:35.470      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_4/ord_x/im/shutdown'
Jul  7 20:10:35.469      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_5/ord_x/im/shutdown'
Jul  7 20:10:35.469      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_6/ord_x/im/shutdown'
Jul  7 20:10:35.469      144      1      4      474      Apply/abort called
'cfg/if/act/GigabitEthernet0_3_4_7/ord_x/im/shutdown'
Jul  7 20:10:35.469      144      1      4      474      Apply/abort batch s
tarted
Jul  7 20:10:35.469      144      1      4      474      verify reply: accep
t
      '___'
Jul  7 20:10:35.469      144      1      4      474      verify reply: accep
t
      '___'
Jul  7 20:10:35.469      144      1      4      474      verify reply: accep
t
      '___'
!
!
!

```

The output shows that changes to the SysDB local plane were verified and accepted.

In the following example, the cfgmgr trace details are displayed.

```
RP/0/RP0/CPU0:router# show cfgmgr trace
```

```

69 wrapping entries (2048 possible, 0 filtered, 69 total)
Jul  5 14:47:17.967 cfgmgr/common 0/RP0/CPU0 t3 Config media returned from disk
util: '/disk0/'.
Jul  5 14:47:46.994 cfgmgr/common 0/RP0/CPU0 t1 Received a state change event.
State is 'active'
Jul  5 14:47:47.218 cfgmgr/common 0/RP0/CPU0 t1 Config media returned from disk
util: '/disk0/'.
Jul  5 14:47:56.502 cfgmgr/common 0/RP0/CPU0 t6 Received a state change event.
State is 'active'
Jul  5 14:47:56.512 cfgmgr/common 0/RP0/CPU0 t4 State of the request queue is '
PROCESSABLE'
Jul  5 14:47:56.520 cfgmgr/common 0/RP0/CPU0 t4 Startup config apply requested
with option '0x2'
Jul  5 14:47:57.471 cfgmgr/common 0/RP0/CPU0 t4 Attempting to apply ascii admin
startup config from file '/qsm/cfsroot/admin/admin.cfg'.
Jul  5 14:48:09.156 cfgmgr/common 0/RP0/CPU0 t4 Clean all admin files since adm

```

```

in commit is empty.
Jul  5 14:48:28.044 cfgmgr/common 0/RP0/CPU0 t6  Infra band COMPLETE 0
Jul  5 14:49:11.832 cfgmgr/common 0/RP0/CPU0 t4  Suspend flag value 1
Jul  5 14:49:11.832 cfgmgr/common 0/RP0/CPU0 t4  State of the request queue is '
NOT-PROCESSABLE'
Jul  5 14:51:22.738 cfgmgr/common 0/RP0/CPU0 t4  Suspend flag value 1
Jul  5 14:51:22.738 cfgmgr/common 0/RP0/CPU0 t4  State of the request queue is '
NOT-PROCESSABLE'
Jul  5 14:51:22.738 cfgmgr/common 0/RP0/CPU0 t4  State of the request queue is '
PROCESSABLE'
Jul  5 14:51:22.738 cfgmgr/common 0/RP0/CPU0 t4  Startup config apply requested
with option '0x1'
Jul  5 14:51:22.793 cfgmgr/common 0/RP0/CPU0 t4  Turboboot flag = '0x0', Passwor
d recovery flag = '0x0'
Jul  5 14:51:26.114 cfgmgr/common 0/RP0/CPU0 t4  Attempting to apply binary LR s
tartup config.
Jul  5 14:51:26.128 cfgmgr/common 0/RP0/CPU0 t4  commitdb_purge_entries called w
ith option '0x0'
Jul  5 14:51:26.135 cfgmgr/common 0/RP0/CPU0 t4  commitdb_check_status returns s
tatus - '0x0' with error: 'No error'
Jul  5 14:51:26.245 cfgmgr/common 0/RP0/CPU0 t4  commitdb_load_changes returns e
rror: 'Invalid argument'
Jul  5 14:51:26.288 cfgmgr/common 0/RP0/CPU0 t4  commitdb_create_delta returns e
rror: 'Invalid argument'
Jul  5 14:51:26.296 cfgmgr/common 0/RP0/CPU0 t4  commitdb_save_running_from_comm
itdb returns error: 'Invalid argument'
!
!
!

```

The output shows that the configuration files are stored on disk0 and the state of the node is active (State is 'active'). The administration configuration is applied during startup for the designated shelf controller (DSC) (Attempting to apply ascii admin startup config from file '/qsm/cfsroot/admin/admin.cfg'). The secure domain router (SDR)-specific configuration is applied from the saved binary check points (Attempting to apply binary LR s tartup config.). Several invalid argument errors were returned when attempting to restore the startup configuration from the binary checkpoints. If there are invalid argument errors, contact Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the Preface.

The output also shows the state of queue as PROCESSABLE or NOT-PROCESSABLE. A NOT-PROCESSABLE state can indicate that the interface is still in the preconfiguration state. If the queue is not processable, then interfaces cannot be restored. This queue state information can be looked at along with the startup config information, as it is one of the gating factors for allowing the queue to be processable.

In the following example, the a list of historical changes to the configuration is displayed.

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

Sno.	Event	Info	Time Stamp
~~~~	~~~~	~~~~	~~~~~
1	commit	id 1000000001	Mon Aug 14 17:09:58 2006
2	commit	id 1000000002	Mon Aug 14 17:41:34 2006
3	commit	id 1000000003	Mon Aug 21 08:05:42 2006
4	commit	id 1000000004	Mon Aug 21 12:39:31 2006
5	commit	id 1000000005	Mon Aug 21 12:43:29 2006
6	commit	id 1000000006	Mon Aug 21 12:45:48 2006
7	commit	id 1000000007	Mon Aug 21 13:23:08 2006
8	commit	id 1000000008	Mon Aug 21 13:24:56 2006
9	commit	id 1000000009	Mon Aug 21 13:26:05 2006
10	commit	id 1000000010	Mon Aug 21 13:55:08 2006
11	commit	id 1000000011	Tue Aug 22 13:52:09 2006

12	commit	id 1000000012	Tue Aug 22 14:15:34 2006
13	commit	id 1000000013	Wed Aug 23 07:26:38 2006
14	commit	id 1000000014	Wed Aug 23 10:44:20 2006
15	commit	id 1000000015	Wed Aug 23 10:44:44 2006
16	commit	id 1000000016	Wed Aug 23 10:44:58 2006
17	commit	id 1000000017	Wed Aug 23 17:47:45 2006
18	commit	id 1000000018	Thu Aug 24 07:55:57 2006
19	commit	id 1000000019	Thu Aug 24 07:58:05 2006
20	commit	id 1000000020	Thu Aug 24 08:01:00 2006
21	commit	id 1000000021	Thu Aug 24 08:01:17 2006
22	commit	id 1000000022	Thu Aug 24 08:01:47 2006
23	commit	id 1000000023	Thu Aug 24 08:02:21 2006
24	commit	id 1000000024	Thu Aug 24 08:03:35 2006
25	commit	id 1000000025	Mon Aug 28 05:34:58 2006
26	commit	id 1000000026	Mon Aug 28 05:50:10 2006
27	commit	id 1000000027	Mon Aug 28 06:07:13 2006
28	commit	id 1000000028	Mon Aug 28 06:09:15 2006
29	commit	id 1000000029	Mon Aug 28 06:44:35 2006
30	commit	id 1000000030	Mon Aug 28 08:36:33 2006
31	commit	id 1000000031	Mon Aug 28 09:40:48 2006
32	commit	id 1000000032	Mon Aug 28 11:29:35 2006
33	commit	id 1000000033	Mon Aug 28 11:54:54 2006
34	commit	id 1000000034	Mon Aug 28 12:29:37 2006
35	commit	id 1000000001	Mon Aug 28 20:28:44 2006
36	commit	id 1000000002	Mon Aug 28 21:07:05 2006

In the following example, detailed information on the last committed configuration is displayed.

```
RP/0/RP0/CPU0:router# show configuration commit changes last 1
```

```
Building configuration...
interface Bundle-Ether28
  description Connected to P2_CRS-8 Bundle-Ether 28
  ipv4 address 10.12.28.1 255.255.255.0
  bundle minimum-active links 1
  bundle minimum-active bandwidth 1000000
!
interface Bundle-Ether28.1
  description Connected to P2_CRS-8 Bundle-Ether 28.1
  ipv4 address 10.12.29.1 255.255.255.0
  dot1q vlan 29
!
interface Bundle-Ether28.2
  description Connected to P2_CRS-8 Bundle-Ether 28.2
  ipv4 address 10.12.30.1 255.255.255.0
  dot1q vlan 30
!
interface Bundle-Ether28.3
  description Connected to P2_CRS-8 Bundle-Ether 28.3
  ipv4 address 10.12.31.1 255.255.255.0
  dot1q vlan 31
!
interface Bundle-POS24
  description Connected to P2_CRS-8 Bundle-POS 24
  ipv4 address 10.12.24.1 255.255.255.0
  bundle minimum-active links 1
  bundle minimum-active bandwidth 2488320
!
no interface Loopback0
interface Loopback0
  ipv4 address 10.1.1.1 255.255.255.255
!

interface MgmtEth0/RP0/CPU0/0
```

```

no description
description Connected to Lab LAN
no ipv4 address 172.29.52.70 255.255.255.0
ipv4 address 172.29.52.70 255.255.255.0
.
.
.
router ospf 100
router-id 10.1.1.1
router-id Loopback0
area 0
interface Loopback0
passive enable
!
interface GigabitEthernet0/1/5/2
!
interface POS0/1/0/1
!
!
!
mpls ldp
router-id Loopback0
log
neighbor
graceful-restart
!
interface POS0/1/0/1
!
interface GigabitEthernet0/1/5/2
!
!
mpls oam
!
ssh server
xml agent tty
xml agent corba
http server
end

```

In the following example, information on any configurations that failed during startup is displayed.

```

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

!!20:16:32 UTC Mon Aug 28 2006
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
domain-lookup
router ospf 100
area 0
interface POS0/1/0/1
mpls ldp
router-id Loopback0
log
neighbor
graceful-restart
interface POS0/1/0/1
interface GigabitEthernet0/1/5/2
mpls oam
router igmp
version 1
ssh server
xml agent tty
xml agent corba
http server

```



In the following example, a check of the current configuration for any missing configurations is run and the results are displayed.

```
RP/0/RP0/CPU0:router# cfs check
```

```
Creating any missing directories in Configuration File system...OK
Initializing Configuration Version Manager...OK
Syncing commit database with running configuration...OK
Re-initializing cache files...OK
Updating Commit Database. Please wait...[OK]
```

## Using the show configuration failed Command

Use the **show configuration failed** command to browse a failed configuration. The configuration can be classified as failed during startup or during a configuration commit.

- [Startup Failed Configuration, page 1-47](#)
- [Commit Configuration Failed, page 1-48](#)

### Startup Failed Configuration

A configuration can be classified as failed during startup for three reasons:

- Syntax errors—syntax errors are generated by the parser and usually indicate that there is an incompatibility with the command-line interface (CLI) commands. Correct the syntax errors and reapply the configuration. A syntax error can be an invalid CLI entry or a CLI syntax change. See the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the [Preface](#) for information on obtaining Cisco IOS XR software CLI documentation.
- Semantic errors—semantic errors are generated by the backend components when the configuration is being restored by the configuration manager during startup of the router. Semantic errors include logical problems (invalid logic).
- Apply errors—apply errors are generated when a configuration has been successfully verified and accepted as part of running configuration but the backend component is not able to update its operational state. The configuration shows both as the running configuration (since it was correctly verified) and as a failed configuration because of the backend operational error. To find the component apply owner, use the describe on the CLI that failed to be applied.



#### Note

You may browse startup failed configurations for up to the previous four router reloads.

Use the **show configuration failed startup** command and the **load configuration failed startup** command to browse and reapply any failed configuration. The **load configuration failed startup** command can be used in configuration mode to load the failed startup configuration into the target configuration session, then the configuration can be modified and committed. See *Cisco IOS XR Getting Started Guide* for information on committing a configuration.

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

```
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
telnet vrf default ipv4
server max-servers 5 interface POS0/7/0/3 router static
address-family ipv4 unicast
0.0.0.0/0 172.18.189.1
```

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

```

router bgp 217
!!% Process did not respond to sysmgr !
RP/0/RP0/CPU0:router#

RP/0/RP0/CPU0:router# config

RP/0/RP0/CPU0:router(config)# load config failed startup noerror

Loading. 263 bytes parsed in 1 sec (259)bytes/sec
RP/0/RP0/CPU0:mike3(config-bgp)#show configuration
Building configuration...
telnet vrf default ipv4 server max-servers 5 router static
address-family ipv4 unicast
    0.0.0.0/0 172.18.189.1
    !
!
router bgp 217
!
end

```

The failed configuration is loaded into the target configuration, minus the errors that caused the startup configuration to fail.

```
RP/0/RP0/CPU0:router(config-bgp)# commit
```

Use the **show configuration failed** command to display failed items in the last configuration commit, including reasons for the error.

In any mode, the configuration failures from the most recent commit operation are displayed.

The **show configuration failed** command can be used in EXEC mode and configuration mode. The command is used in EXEC mode when the configuration does not load during startup. The command is used in configuration mode to display information when a commit fails.

The following example shows the **show configuration failed** command.

```

RP/0/RP0/CPU0:router(config)# interface pos 0/6/0/4
RP/0/RP0/CPU0:router(config-if)# no vrf
RP/0/RP0/CPU0:router(config-if)# commit

% Failed to commit one or more configuration items during an atomic operation, no changes
have been made. Please use 'show configuration failed' to view the errors

RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# show configuration failed

Wed May  2 13:14:08.426 EST EDT
!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS interface POS0/6/0/4 no vrf !!
% The interface's numbered and unnumbered IPv4/IPv6 addresses must be removed prior to
changing or deleting the VRF !

```



#### Note

The **show configuration failed** command in configuration mode only exists as long as the configuration session is active. Once you exit configuration mode, the command cannot be used to display the failed configuration.

## Commit Configuration Failed

The following example shows an invalid task ID configuration that fails to commit. The **show configuration failed** command provides information on why the configuration failed.

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

```

RP/0/RP0/CPU0:router(config-tg)# commit

% Failed to commit one or more configuration items during an atomic operation, s

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

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

If a configuration commit fails, do not exit configuration mode (return to EXEC mode) as you will not
be able to view the failed configuration.

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup bgp
RP/0/RP0/CPU0:router(config-tg)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:y

% Failed to commit one or more configuration items during an atomic operation, s

RP/0/RP0/CPU0:router(config)# exit
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:n
RP/0/RP0/CPU0:router# show configuration failed
RP/0/RP0/CPU0:router#

```

## ASIC Errors

The following ASIC error types are supported:

- FIA (Fabric Interface ASIC)
- PSE (packet switching engine)
- Cpuctrl
- Egressq
- Fabricq
- Discovery
- Plim-asic

The following ASIC error classifications are supported:

- Single Bit Errors (SBE)—Correctable ECC protected single bit errors in external or internal memory

Not reported to PM on each occurrence and reported to the platform manager (PM) as Minor when software threshold rate is exceeded. Report alarm using Alarm Logging, and Debugging Event Management System (ALDEMS).

Error data:

- Address—Address that encountered the SBE
- Syndrome—Syndrome if available

- Multiple Bit Errors—Uncorrectable multiple bit error in memory.

Reported to PM as Major and ALDEMS for each occurrence.

Error data:

- Address—Address that encountered the SBE
  - Data—Actual error data
  - PARITY Errors—Parity error in all applicable memory  
Reported to PM as Major.
  - Cyclic redundancy check (CRC) Errors—CRC errors in EIO other links.  
Not reported for each occurrence. When the threshold is reached it is reported as Major to the PM.
  - GENERIC Errors—Errors that do not fall under any of the other classifications.  
Threshold and alarm reporting is done.
  - RESET Errors—Logged for each reset instance of the ASIC.  
Reported to PM when threshold is exceeded.
- Error data:

- Interrupt status—Interrupt status bits due to ASIC reset.
- Halt status—Halt status bits.
- Reset node key—Key for the error node that causes the reset.
- Time—Reset time.

The following ASIC error fault severities are supported:

- Critical—Affected component is unusable or card is reset if no redundant card exists
- Major—Partially service affecting fault or if redundant card, do a failover otherwise the card runs in degraded mode
- Minor—Non-service affecting fault
- OK—No fault

Use the **show asic-errors** command to view if any ASIC errors have occurred on nodes. The following example shows the output of the **show asic-errors** command.

```
RP/0/RP0/CPU0:router# show asic-errors plim asic 0 all location 0/0/CPU0
```

```
Wed May  2 13:45:00.626 EST EDT
*****
*                               *
*               Single Bit Errors               *
*                               *
*****
*                               *
*               Multiple Bit Errors               *
*                               *
*****
*                               *
*               Parity Errors                     *
*                               *
*****
*                               *
*               CRC Errors                       *
*                               *
*****
*                               *
*               Generic Errors                   *
*                               *
*****
Name           : Port 0 PAR XGM link fault
Node Key       : 0x105051d
Thresh/period(s): 0/0    Alarm state: OFF
Error count    : 3703662240
Last clearing   : Thu Mar 29 11:59:32 2007
Last N errors   : 50
-----
First N errors.
```

```

@Time, Error-Data
-----
Mar 29 11:59:32.186: PAR XGM link fault
Mar 31 10:37:52.635: PAR XGM link fault
Mar 31 10:37:53.646: PAR XGM link fault
Mar 31 10:37:54.656: PAR XGM link fault
Mar 31 10:37:55.666: PAR XGM link fault
Mar 31 10:37:56.676: PAR XGM link fault
Mar 31 10:37:57.686: PAR XGM link fault
Mar 31 10:37:58.696: PAR XGM link fault
Mar 31 10:37:59.706: PAR XGM link fault
Mar 31 10:38:00.716: PAR XGM link fault
Mar 31 10:38:01.726: PAR XGM link fault
Mar 31 10:38:02.736: PAR XGM link fault
Mar 31 10:38:03.746: PAR XGM link fault
Mar 31 10:38:04.756: PAR XGM link fault
Mar 31 10:38:05.766: PAR XGM link fault
Mar 31 10:38:06.776: PAR XGM link fault
Mar 31 10:38:07.786: PAR XGM link fault
Mar 31 10:38:08.351: PAR XGM link fault
Mar 31 10:38:08.796: PAR XGM link fault
Mar 31 10:38:09.806: PAR XGM link fault
Mar 31 10:38:10.816: PAR XGM link fault
Mar 31 10:38:11.826: PAR XGM link fault
Mar 31 10:38:12.836: PAR XGM link fault
Mar 31 10:38:13.846: PAR XGM link fault
Mar 31 10:38:14.856: PAR XGM link fault
Last N errors.
@Time, Error-Data
-----
Apr 5 05:48:14.297: PAR XGM link fault
Apr 5 05:48:15.307: PAR XGM link fault
Apr 5 05:48:16.317: PAR XGM link fault
Apr 5 05:48:17.327: PAR XGM link fault
Apr 5 05:48:18.337: PAR XGM link fault
Apr 5 05:48:19.347: PAR XGM link fault
Apr 5 05:48:20.357: PAR XGM link fault
Apr 5 05:48:21.367: PAR XGM link fault
Apr 5 05:48:22.377: PAR XGM link fault
Apr 5 05:48:23.387: PAR XGM link fault
Apr 5 05:48:24.398: PAR XGM link fault
Apr 5 05:48:25.408: PAR XGM link fault
Apr 5 05:48:26.418: PAR XGM link fault
Apr 5 05:48:27.428: PAR XGM link fault
Apr 5 05:48:28.438: PAR XGM link fault
Apr 5 05:48:29.180: PAR XGM link fault
Apr 5 05:48:29.448: PAR XGM link fault
Apr 5 05:48:30.459: PAR XGM link fault
Apr 5 05:48:31.469: PAR XGM link fault
Apr 5 05:48:32.479: PAR XGM link fault
Apr 5 05:48:33.489: PAR XGM link fault
Apr 5 05:48:34.499: PAR XGM link fault
Apr 5 05:48:35.509: PAR XGM link fault
Apr 5 05:48:36.519: PAR XGM link fault
Apr 5 05:48:37.529: PAR XGM link fault
-----
Name          : Port 3 PAR XGM link fault
Node Key      : 0x105081d
Thresh/period(s): 0/0    Alarm state: OFF
Error count   : 332349486
Last clearing  : Mon Apr 16 10:44:39 2007
Last N errors  : 21
-----
First N errors.

```

```

@Time, Error-Data
-----
Apr 16 10:44:39.245: PAR XGM link fault
Apr 16 10:44:40.255: PAR XGM link fault
Apr 16 10:44:41.265: PAR XGM link fault
Apr 16 10:44:42.275: PAR XGM link fault
Apr 16 10:44:43.285: PAR XGM link fault
Apr 16 10:44:44.295: PAR XGM link fault
Apr 16 10:44:45.305: PAR XGM link fault
Apr 16 10:44:45.487: PAR XGM link fault
Apr 16 10:44:46.315: PAR XGM link fault
Apr 16 10:44:47.325: PAR XGM link fault
Apr 16 10:44:48.335: PAR XGM link fault
Apr 16 10:44:49.345: PAR XGM link fault
Apr 16 10:44:50.355: PAR XGM link fault
Apr 16 10:44:51.365: PAR XGM link fault
Apr 16 10:44:52.375: PAR XGM link fault
Apr 16 10:44:53.385: PAR XGM link fault
Apr 16 10:44:54.395: PAR XGM link fault
Apr 16 10:44:55.405: PAR XGM link fault
Apr 16 10:44:56.415: PAR XGM link fault
Apr 16 10:44:57.425: PAR XGM link fault
Apr 16 10:54:23.147: PAR XGM link fault
-----
Name           : MBP BP bad pattern
Node Key       : 0x205030f
Thresh/period(s): 100/5  Alarm state: OFF
Error count    : 1
Last clearing   : Thu Mar 29 11:56:32 2007
Last N errors  : 1
-----
First N errors.
@Time, Error-Data
-----
Mar 29 11:56:32.749: MBP BP bad pattern
-----
*****
*                               ASIC Reset Errors                               *
*****

```

The following example shows how to display ASIC errors. The ASIC-ERRORs folder is created after the first node reset. A folder is created for each node that has reloaded because of an ASIC error. If the ASIC-ERROR folder does not exist, there have not been any node resets on the system.

```
RP/0/RP0/CPU0:router# dir harddisk:
```

```
Directory of harddisk:
```

```

5          drwx  4096          Fri Jun 10 10:27:32 2005  LOST.DIR
6          drwx  4096          Fri Jun 10 10:27:32 2005  usr
7          drwx  4096          Fri Jun 10 10:27:32 2005  var
131328     -rwx 173056          Tue Apr 18 17:18:50 2006  instdb_backup.tar
19         drwx  4096          Tue Apr 18 17:32:46 2006  dumper
1880       drwx  4096          Fri Oct 14 13:52:22 2005  ASIC-ERROR

```

The following example shows how to display node-specific ASIC errors.

```
RP/0/RP0/CPU0:router# dir harddisk:/ASIC-ERROR
```

```
Directory of harddisk:/ASIC-ERROR
```

```
1881       drwx  4096          Thu Jun 16 08:32:14 2005  node0_3_CPU0
```

```
2141          drwx  4096          Fri Oct 14 13:52:22 2005  node0_5_CPU0
```

The output shows that there were two line card reloads caused by ASIC errors (June 16 and October 14).

The following example lists the PSE files for a specific node reload. The PSE files contain the actual ASIC error data that triggered the reload.

```
RP/0/RP0/CPU0:router# dir harddisk:/ASIC-ERROR/node0_3_CPU0
```

```
Directory of harddisk:/ASIC-ERROR/node0_3_CPU0
```

```
123273312    -rwx  4823          Sat Aug 13 20:04:06 2005  pse_00.err
123273376    -rwx  4794          Sat Aug 13 20:04:06 2005  pse_01.err
```

The following example shows how to display the contents of a specific PSE file.

```
RP/0/RP0/CPU0:router# more harddisk:/ASIC-ERROR/node0_3_CPU0/pse_00.err
```

```
Next file write offset = 4823
```

```
^@
##### Start of data pse_00_061605_083214.err #####
*****
*                ASIC Errors Summary                *
*****
Number of nodes      : 2
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count   : 0
```

The following example shows how to display a summary for each ASIC. If an error is displayed, dump the individual asic instance number to obtain details on the ASIC error.

```
RP/0/RP0/CPU0:router# show asic-errors all location 0/6/cpu0
```

```
*****
*                Fia ASIC Error Summary                *
*****
Instance             : 0
Number of nodes       : 130
SBE error count       : 0
MBE error count       : 0
Parity error count    : 0
CRC error count       : 0
Generic error count   : 0
Reset error count     : 0
-----
Instance             : 1
Number of nodes       : 130
SBE error count       : 0
MBE error count       : 0
Parity error count    : 0
CRC error count       : 0
Generic error count   : 0
Reset error count     : 0
-----
*****
*                Pse ASIC Error Summary                *
*****
Instance             : 0
Number of nodes       : 2
SBE error count       : 0
```

```

MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```

```

Instance             : 1
Number of nodes      : 2
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```

```

*****
*                               Cpuctrl ASIC Error Summary                               *
*****

```

```

Instance             : 0
Number of nodes      : 0
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```

```

*****
*                               Egressq ASIC Error Summary                               *
*****

```

```

Instance             : 0
Number of nodes      : 1
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```

```

*****
*                               Fabricq ASIC Error Summary                               *
*****

```

```

Instance             : 0
Number of nodes      : 3
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```

```

Instance             : 1
Number of nodes      : 2
SBE error count      : 0
MBE error count      : 0
Parity error count   : 0
CRC error count      : 0
Generic error count  : 0
Reset error count    : 0
-----

```



```

*****
*                               Ingressq ASIC Error Summary                               *
*****
Instance           : 0
Number of nodes    : 1
SBE error count    : 0
MBE error count    : 0
Parity error count : 0
CRC error count    : 0
Generic error count : 0
Reset error count  : 0
-----

```

```

*****
*                               Discovery ASIC Error Summary                               *
*****
Instance           : 0
Number of nodes    : 0
SBE error count    : 0
MBE error count    : 0
Parity error count : 0
CRC error count    : 0
Generic error count : 0
Reset error count  : 0
-----

```

```

*****
*                               Plim-asic ASIC Error Summary                               *
*****
Instance           : 0
Number of nodes    : 2
SBE error count    : 0
MBE error count    : 0
Parity error count : 0
CRC error count    : 0
Generic error count : 2
Reset error count  : 0
-----
Instance           : 1
Number of nodes    : 3
SBE error count    : 0
MBE error count    : 0
Parity error count : 0
CRC error count    : 0
Generic error count : 22448
Reset error count  : 0
-----

```

RP/0/RP0/CPU0:router# **show controllers asic sharq instance help-instance location 0/0/CPU0**

```

Wed May  2 13:47:22.838 EST EDT Total 1 Instance
Instance Number --- Instance Name
      0              SHARQ

```

RP/0/RP0/CPU0:router# **show controllers asic metro instance help-instance location 0/0/CPU0**

```

Wed May  2 13:47:51.817 EST EDT Total 2 Instance
Instance Number --- Instance Name
      0              IngressPSE

```

1

EgressPSE

## Trace Commands

Trace commands provide an ‘always on’ debug feature. Many major functions in Cisco IOS XR software have “trace” functionality to show the last actions it conducted allowing you to analyze function events. Use the show trace commands to display the trace data for a specific feature or process. Use the ? in the CLI to determine if a command has the **trace** keyword. The following example shows that the **show arp** command has the **trace** keyword.

```
RP/0/RP0/CPU0:router# show arp ?
```

```
A.B.C.D          IP address or hostname of ARP entry
Bundle-Ether     Aggregated Ethernet interface(s)
GigabitEthernet  GigabitEthernet/IEEE 802.3 interface(s)
H.H.H           48-bit hardware address of ARP entry
MgmtEth          Ethernet/IEEE 802.3 interface(s)
idb              Show the internal ARP interface data block
location         specify a node name
trace            Show trace data for the ARP component
traffic          ARP traffic statistics
vrf              Specify a VRF
|               Output Modifiers
<cr>
```

The following example shows the last 20 events in the address resolution protocol (ARP) table.

```
RP/0/RP0/CPU0:router# show arp trace tailf last 20
```

```
1349 wrapping entries (2048 possible, 0 filtered, 1349 total)
Apr 19 09:52:29.857 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: creating incomplete entry for
address: 172.18.105.255
Apr 19 09:52:34.501 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255
Apr 19 09:52:41.856 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
Apr 19 09:52:46.324 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255
Apr 19 09:52:59.979 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: entry 172.18.105.255: deleted
from table
Apr 19 09:59:37.463 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
Apr 19 09:59:37.463 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: creating incomplete entry for
address: 172.18.105.255
Apr 19 09:59:39.515 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
Apr 19 09:59:42.082 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255
Apr 19 09:59:45.007 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: entry 172.18.105.255: deleted
from table
Apr 19 09:59:50.101 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
Apr 19 09:59:50.101 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: creating incomplete entry for
address: 172.18.105.255
Apr 19 09:59:54.820 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255
Apr 19 10:00:00.008 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: entry 172.18.105.255: deleted
from table
Apr 19 10:04:11.675 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
```

```

Apr 19 10:04:11.675 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: creating incomplete entry for
address: 172.18.105.255
Apr 19 10:04:16.272 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255
Apr 19 10:04:30.028 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: entry 172.18.105.255: deleted
from table
Apr 19 10:04:44.097 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: received address resolution
request for 172.18.105.255
Apr 19 10:04:44.097 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: creating incomplete entry for
address: 172.18.105.255
Apr 19 10:04:48.810 ipv4_arp/arp 0/RP0/CPU0 t1 ARP-TABLE: address resolution failed for
172.18.105.255

```

## Packets

By default, if packet capture is not enabled on an interface, the **show packet-memory** command displays only punt packets and software switch packets (router generated packets or anything that impacts the line card CPU). You have the option of retrieving information on packets that are switched and punted in the software using the **show captured packets** command. You have to turn on packet capture using the **capture software packets** command, then use the **show captured packets** command to display the packet capture content.

The following example shows that interface does not have packet capture enabled using the **capture software packets** command:

```
RP/0/RP0/CPU0:router# show captured packets ingress interface pos 0/1/0/0 location
0/RP0/CPU0
```

```

please enable packet capture on interface to see pkts
RP/0/RP0/CPU0:router#

```



### Note

This feature is supported on the Cisco CRS-1.

To turn on packet capture and view capture packet output, perform the following procedure.

### SUMMARY STEPS

1. **configure**
2. **interface** *type instance*
3. **capture software packets**
4. **end**  
or  
**commit**
5. **show captured packets {ingress | egress} [interface type instance] [hexdump] [last number]**  
**[single-line] location node-id**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>  <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 1	<b>interface</b> <i>type instance</i>  <b>Example:</b> RP/0/RP0/CPU0:router(config)# interface pos 0/1/0/0	Enters interface configuration mode.
Step 2	<b>capture software packets</b>  <b>Example:</b> RP/0/RP0/CPU0:router(config-if)# capture software packets	Turns on software packet capture for the POS 01/0/0 interface.
Step 3	<b>end</b> OR <b>commit</b>  <b>Example:</b> RP/0/RP0/CPU0:router(config-if)# end OR RP/0/RP0/CPU0:router(config-if)# commit	Saves configuration changes. <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes:  Uncommitted changes found, commit them before exiting (yes/no/cancel)?  [cancel]: <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
Step 4	<b>show captured packets</b> { <i>ingress</i>   <i>egress</i> } [ <i>hexdump</i> ] [ <i>interface type instance</i> ] [ <i>last number</i> ] [ <i>single-line</i> ] <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show captured packets ingress location cpu 0/1/cpu0	Displays information on packets that are switched and punted in the software.

The following example shows how to turn on packet capture for POS 0/1/0/0.

```
RP/0/RP0/CPU0:router(config)# interface pos 0/1/0/0
RP/0/RP0/CPU0:router(config-if)# capture software packets
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
```

The following example shows the output of the **show captured packets** command for POS 0/6/0/4.

```
RP/0/RP0/CPU0:router# show captured packets ingress interface pos 0/6/0/4 location
0/6/cpu0

Wed May  2 13:55:50.748 EST EDT
-----
packets captured on interface in ingress direction buffer overflow pkt drops:55050,
current: 200, non wrapping: 0
maximum: 200
-----
Wrapping entries
-----
[1] Apr 25 05:40:00.305, len: 60, hits: 1, i/p i/f: POS0/6/0/4
    [punt reason: IPV4_NO_MATCH]
    [HDLC: flags 0x0f00 type 0x0800]
    [IPV4: source 172.19.90.6, dest 172.19.90.2 ihl 5, ver 4, tos 0
      id 0, len 56, prot 61, ttl 64, sum 6e5a, offset 0]
    00000000 00000000 00000000 00000000 00100fff 00240306 0d383570 9c106775
    ba210000

[2] Apr 25 05:40:00.306, len: 60, hits: 1, i/p i/f: POS0/6/0/4
    [punt reason: IPV4_NO_MATCH]
    [HDLC: flags 0x0f00 type 0x0800]
    [IPV4: source 172.19.90.6, dest 172.19.90.2 ihl 5, ver 4, tos 0
      id 0, len 56, prot 61, ttl 64, sum 6e5a, offset 0]
    00000000 00000000 00000000 00000000 00100fff 00240332 0d39aa68 6e6ab1c5
    12cb0000

[3] Apr 25 05:40:00.307, len: 60, hits: 1, i/p i/f: POS0/6/0/4
    [punt reason: IPV4_NO_MATCH]
    [HDLC: flags 0x0f00 type 0x0800]
    [IPV4: source 172.19.90.6, dest 172.19.90.2 ihl 5, ver 4, tos 0
      id 0, len 56, prot 61, ttl 64, sum 6e5a, offset 0]
    00000000 00000000 00000000 00000000 00100fff 0024035c 0d3b0e7a 897fe5d9
    32b80000

[4] Apr 25 05:40:01.308, len: 60, hits: 1, i/p i/f: POS0/6/0/4
    [punt reason: IPV4_NO_MATCH]
    [HDLC: flags 0x0f00 type 0x0800]
    [IPV4: source 172.19.90.6, dest 172.19.90.2 ihl 5, ver 4, tos 0
      id 0, len 56, prot 61, ttl 64, sum 6e5a, offset 0]
    00000000 00000000 00000000 00000000 00100fff 00240386 0d3c727e 17fc089b b82f0000

[5] Apr 25 05:40:01.309, len: 60, hits: 1, i/p i/f: POS0/6/0/4
    [punt reason: IPV4_NO_MATCH]
    [HDLC: flags 0x0f00 type 0x0800]
    [IPV4: source 172.19.90.6, dest 172.19.90.2 ihl 5, ver 4, tos 0
      id 0, len 56, prot 61, ttl 64, sum 6e5a, offset 0]
    00000000 00000000 00000000 00000000 00100fff 002403b2 0d3de784 3bed359f a39c0000
.
.
.
```

## Logging Archive for Harddisk

Use the **logging archive** command to configure attributes for archiving syslogs. Configuring the logging archive is recommended as sometimes syslog does not make it over the network and the archive can be used for post problem analysis help.

The following example shows how to configure a syslog logging archive that uses the harddisk, is for all severities (0 through 7), and collects logs daily.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# logging archive
RP/0/RP0/CPU0:router(config-logging-arch)# device harddisk
RP/0/RP0/CPU0:router(config-logging-arch)# severity debugging
RP/0/RP0/CPU0:router(config-logging-arch)# frequency daily
```

**Note**

Harddisk logging is not recommended for normal operation. You must enable this under the direction of the Cisco Support team.

## SNMP Polling Awareness of SystemOwner, LR Owner, MIB Location

**Note**

If you are experiencing timeouts of the SNMP process, see the troubleshooting information in the [“Troubleshooting SNMP Timeouts”](#) section on page 8-194.

By default, if you configure SNMP in the Secure Domain Router (SDR), you only see what is in the logical router (LR plane) and you do not have snmp access to fan, power, and fabric card information (admin plane). If you add systemowner on the community string using the **snmp-server community** command, you will have access to the entire system allowing you to poll information such as fabric information and status.

In order to view entire MIB table, the community string needs to have ‘systemowner’. This allows the user to view admin plane objects as well as LR plane.

To locate and download MIBs, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu:

<http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

## Error File Locations and Data Collection Scripts

Errors are logged and stored on the system.

Data collection scripts are used to collect relevant information for troubleshooting the system. Scripts are stored locally. The following script types are supported:

- [Error File Locations](#), page 1-61
- [Sysmgr Collection Scripts](#), page 1-62
- [Wdsysmon Collection Scripts](#), page 1-63
- [Shutdown Collection Scripts](#), page 1-66

- [ASIC error Collection Scripts, page 1-66](#)

## Error File Locations

Error files are located in the following folders:

- [harddisk:, page 1-61](#)
- [Default disk location, page 1-62](#)

Use the **dir** command to display a list of files on a file system or in a specific directory. The following examples show how to display the contents of the harddisk: directory, and the files in the var and shutdown directories.

```
RP/0/RP0/CPU0:router# dir harddisk:
```

```
Directory of harddisk:
```

```

5          drwx  4096          Tue Oct  4 09:55:32 2005  LOST.DIR
6          drwx  4096          Tue Oct  4 09:55:34 2005  usr
7          drwx  4096          Tue Oct  4 09:55:18 2005  var
131328     -rwx  31744         Fri Apr 28 17:38:34 2006  instdb_backup.tar
15         drwx  4096          Wed Apr  5 18:27:48 2006  dumper
3678       drwx  4096          Wed Mar 15 17:03:48 2006  shutdown
4008       drwx  4096          Thu Mar  2 09:24:10 2006  malloc_dump
```

```
39929724928 bytes total (39908044800 bytes free)
```

```
RP/0/RP0/CPU0:router# dir harddisk:/var
```

```
Directory of harddisk:/var
```

```

8          drwx  4096          Tue Oct  4 09:55:18 2005  log
9          drwx  4096          Tue Oct  4 09:55:18 2005  tmp
```

```
39929724928 bytes total (39908044800 bytes free)
```

Use the **cd** command to change the current working directory.

The following example shows how to change from the default working directory to the harddisk: directory. The **pwd** command displays the present working directory.

```
RP/0/RP0/CPU0:router# cd
```

```
RP/0/RP0/CPU0:router# pwd
```

```
disk0:/usr
```

```
RP/0/RP0/CPU0:router# cd harddisk:
```

```
RP/0/RP0/CPU0:router# pwd
```

```
harddisk:
```

Use the **more** command to view the contents of a file.

### harddisk:

The following folders are located on the harddisk:

- shutdown—Contains shutdown scripts collected for nodes
- var/log—Contains any syslog archive data to harddisk if enabled

- ASIC-ERROR—Contains ASIC error data that resulted in a node reload
- asic_snapshot—Contains ASIC data collected in the ASIC error trigger node reload event
- dumper—Contains all process core files in event process crashes
- malloc_dump—Used for memory compare tool

The following example shows the contents of the `hddisk:` folder:

```
RP/0/RP0/CPU0:router# dir hddisk:
```

Directory of hddisk:

```

5          drwx  4096      Fri Jun 10 10:27:32 2005  LOST.DIR
6          drwx  4096      Fri Jun 10 10:27:32 2005  usr
7          drwx  4096      Fri Jun 10 10:27:32 2005  var
19         drwx  4096      Tue Apr 18 17:32:46 2006  dumper
1880       drwx  4096      Fri Oct 14 13:52:22 2005  ASIC-ERROR
1884       drwx  4096      Mon Apr 10 07:36:04 2006  shutdown
1901       drwx  4096      Sat Aug 13 20:04:58 2005  asic_snapshots
6623       drwx  4096      Thu Jul 7 07:12:06 2005  malloc_dump
4460       drwx  4096      Wed Jan 25 09:24:00 2006  pm
```

## Default disk location

The following folders are located on default disk locations such as `disk0:` and `disk1:`

- `wdsysmon_debug`—Contains `wdsysmon` debug data collected in event script is run

The following example shows the contents of the `disk1:` folder. See [“Wdsysmon Collection Scripts” section on page 1-63](#) for information on the contents of the `wdsysmon_debug` folder.

```
RP/0/RP0/CPU0:router# dir disk1:
```

Directory of disk1:

```

2          drwx  16384     Mon Nov 21 13:28:56 2005  LOST.DIR
5          dr-x   16384     Tue Mar 7 17:54:24 2006  bcm-prev
13         dr-x   16384     Tue Mar 7 18:10:19 2006  bcm-cur
3          drwx  16384     Wed Dec 21 21:57:14 2005  wdsysmon_debug
65888      -rwx   54104355   Wed Nov 23 17:27:08 2005  comp-hfr-mini.pie-3.2.2
65984      -rwx   1360942    Tue Mar 7 18:18:15 2006  hfr-k9sec-p.pie-3.3.86.I
66080      -rwx   9215019    Tue Mar 7 18:19:06 2006  hfr-mgbl-p.pie-3.3.86.1I
66176      -rwx   82022548   Tue Mar 7 18:26:20 2006  comp-hfr-mini.vm-3.3.86I
66496      -rwx   31232      Tue Mar 7 20:56:20 2006  instdb_backup.tar
66688      -rwx   3769071    Wed Nov 23 17:28:22 2005  hfr-diags-p.pie-3.2.2
66784      -rwx   1357245    Wed Nov 23 17:29:30 2005  hfr-k9sec-p.pie-3.2.2
66880      -rwx   2440530    Wed Nov 23 17:30:28 2005  hfr-mcast-p.pie-3.2.2
66976      -rwx   7648034    Wed Nov 23 17:31:34 2005  hfr-mgbl-p.pie-3.2.2
67072      -rwx   2414096    Wed Nov 23 17:32:34 2005  hfr-mpls-p.pie-3.2.2
```

```
1024655360 bytes total (859799552 bytes free)
```

## Sysmgr Collection Scripts

The `sysmgr` is responsible for starting, monitoring, stopping, and if necessary, restarting most processes on the system.



Significant sysmgr events are stored in /tmp/sysmgr.log. The log wraps so it is recommended that you save a snapshot to disk at the start of the session. The debug script contains commands that provide a snapshot of the box and also provides details on the specific process having problems.

The following example shows the output of the sysmgr collection script.

```
RP/0/RP0/CPU0:router# run more /tmp/sysmgr.log

Wed May  2 14:03:42.658 EST EDT
01/01 00:00:02.599 1 *** sysmgr_lite spawned***
01/01 00:00:02.600 1 sysmgr_lite: [/dev/sysmgr_shmem] doesn't exist, error=[No such file
or directory]
01/01 00:00:02.600 1 sysmgr_lite: Cold-Reload
01/01 00:00:02.650 1 Hello from init !!
01/01 00:00:02.650 1 Wait for file access through pkgfs
01/01 00:00:03.672 1 Boot Device = disk0:
01/01 00:00:03.672 1 Create event manager
01/01 00:00:03.734 1 Attach to msg channel
01/01 00:00:03.735 1 Create msg handling thread
01/01 00:00:03.735 2 sysmgr_lite_process_msg: In sysmgr_process_msg thread
01/01 00:00:03.735 2 Attaching respawn handler
01/01 00:00:03.736 1 read_init_startup_list: opening directory /pkg/ init.d for .init
files
01/01 00:00:03.736 2 Attaching async handler
01/01 00:00:03.737 2 Attaching sync handler
01/01 00:00:03.737 2 starting ih_timer
01/01 00:00:03.737 2 lite_set_timer: id=1, 1800 seconds
01/01 00:00:03.737 2 Servicing msgs
01/01 00:00:03.738 1 read_init_startup_list: Opening /pkg/init.d/ attach_server.init
01/01 00:00:03.739 1 read_init_startup_list: finished /pkg/init.d/ attach_server.init
pcb->name=attach_server
01/01 00:00:03.739 1 read_init_startup_list: Opening /pkg/init.d/ attachd.init
01/01 00:00:03.739 1 read_init_startup_list: finished /pkg/init.d/ attachd.init
pcb->name=attachd
01/01 00:00:03.739 1 read_init_startup_list: Opening /pkg/init.d/ bcm.init
01/01 00:00:03.739 1 read_init_startup_list: finished /pkg/init.d/ bcm.init
pcb->name=bcm_process
01/01 00:00:03.740 1 read_init_startup_list: Opening /pkg/init.d/ bcm_logger.init
01/01 00:00:03.740 1 read_init_startup_list: finished /pkg/init.d/ bcm_logger.init
pcb->name=bcm_logger
```

## Wdsysmon Collection Scripts

Wdsysmon is the WatchDog and SYStem MONitor. Wdsysmon monitors memory and CPU resources, watches for IP communications and mutual exclusion object (mutex) deadlocks, issues notifications or alarms when resource thresholds are exceeded, and logs historical process data.

The wdsysmon collection scripts contain wdsysmon debug data. The wdsysmon monitors CPU utilization, disk space, and memory, and generates an alarm when a threshold is crossed. The output contains trace information for process resource utilization.

```
RP/0/RP0/CPU0:router# dir disk1:/wdsysmon_debug

Directory of disk1:/wdsysmon_debug

196736      -rwx   37151      Wed Dec 21 17:47:40 2005  debug_evm.364641
196832      -rwx   39422      Wed Dec 21 21:42:10 2005  debug_evm.364640
196928      -rwx   39577      Wed Dec 21 21:57:14 2005  debug_evm.307296
```

1024655360 bytes total (927186944 bytes free)

RP/0/RP0/CPU0:router# **more debug_evm.364641**

/pkg/bin/wdsysmon_debug_evm_script invoked by pid 45103 (wdsysmon) for pid 8200.

Called by wd_heartbeat_timeout_hndlr at line 390 at 17:47:40.029 UTC Wed Dec 21 2005.

-----  
Output from pidin:

pid	tid	name	prio	STATE	Blocked
1	1	procnto	0f	READY	
1	2	procnto	63r	RECEIVE	1
1	3	procnto	63r	RECEIVE	1
1	5	procnto	63r	RECEIVE	1
1	6	procnto	63r	RECEIVE	1
1	7	procnto	63r	RECEIVE	1
1	8	procnto	63r	RECEIVE	1
1	9	procnto	63r	RECEIVE	1
1	10	procnto	63r	RECEIVE	1
1	11	procnto	63r	RECEIVE	1
1	12	procnto	10r	RECEIVE	1
1	13	procnto	10r	RECEIVE	1
1	14	procnto	10r	RECEIVE	1
1	15	procnto	10r	RECEIVE	1
1	16	procnto	10r	RECEIVE	1
1	17	procnto	10r	RECEIVE	1
1	18	procnto	10r	RECEIVE	1
1	19	procnto	10r	RECEIVE	1
1	20	procnto	11r	RECEIVE	1

.  
.  
.

86112	3	pkg/bin/instdir	10r	RECEIVE	1
86112	4	pkg/bin/instdir	10r	CONDVAR	482f9fd8
364641	1	pkg/bin/ksh	10r	SIGSUSPEND	
381026	1	pkg/bin/ksh	10r	SIGSUSPEND	
389219	1	pkg/bin/pidin	10r	REPLY	1

-----  
Output from attach_process -A -p 8200 -i 1

Attaching to process pid = 8200 (pkg/bin/devc-conaux)

No tid specified, following all threads

DLL Loaded by this process

-----

DLL path	Text addr.	Text size	Data addr.	Data size	Version
/pkg/lib/libsysmgr.dll	0xfc122000	0x0000df88	0xfc0c2b14	0x000004ac	0
/pkg/lib/libcerrno.dll	0xfc130000	0x00002f24	0xfc133000	0x00000128	0
/pkg/lib/libcerr_dll_tbl.dll	0xfc134000	0x00004964	0xfc133128	0x00000148	0
/pkg/lib/libltrace.dll	0xfc139000	0x00007adc	0xfc133270	0x00000148	0
/pkg/lib/libinfra.dll	0xfc141000	0x000341a4	0xfc1333b8	0x00000bbc	0
/pkg/lib/libcerrno/libinfra_error.dll	0xfc1121dc	0x00000cd8	0xfc176000	0x000000a8	0
/pkg/lib/libbios.dll	0xfc177000	0x0002dc38	0xfc1a5000	0x00002000	0
/pkg/lib/libcerrno/libevent_manager_error.dll	0xfc1a7000	0x00000e88	0xfc133f74	0x00000088	0
/pkg/lib/libc.dll	0xfc1a8000	0x00079d70	0xfc222000	0x00002000	0
.					
.					
.					

```

/pkg/lib/cerrno/libsysdb_error_callback.dll 0xfc4f3000 0x0000168c 0xfc47ece8 0x0
0000088 0
/pkg/lib/cerrno/libsysdb_error_distrib.dll 0xfc4f5000 0x00001780 0xfc47ed70 0x00
000088 0

```

```

Iteration 1 of 1
-----REPLY (node node0_RP0_CPU0, pid 81994)
-----

```

Current process = "pkg/bin/devc-conaux", PID = 8200 TID = 1

```

trace_back: #0 0xfc1642b8 [MsgSendv]
trace_back: #1 0xfc14e358 [msg_sendv]
trace_back: #2 0xfc49d870 [sysdb_lib_send_opt_v]
trace_back: #3 0xfc4b86fc [sysdb_lib_notification_send_reg]
trace_back: #4 0xfc4b8a7c [sysdb_notification_register_internal]
trace_back: #5 0xfc4b8e74 [_sysdb_register_notification]
trace_back: #6 0xfc274eec [tty_sysdb_cached_item_notify]
trace_back: #7 0xfc275278 [tty_sysdb_cached_items_open]
trace_back: #8 0xfc156b90 [event_conn_evm_handler]
trace_back: #9 0xfc1563ec [event_conn_timeout]
trace_back: #10 0xfc152908 [evm_timeout]
trace_back: #11 0xfc153954 [_event_pulse_handler]
trace_back: #12 0xfc151e94 [event_dispatch]
trace_back: #13 0xfc26c5d8 [tty_io_devctl]
trace_back: #14 0xfc26d7ec [tty_server_main]
trace_back: #15 0x482000b0 [<N/A>]

```

ENDOFSTACKTRACE

Current process = "pkg/bin/devc-conaux", PID = 8200 TID = 2

```

trace_back: #0 0xfc1d4048 [SignalWaitinfo]
trace_back: #1 0xfc1b7d40 [sigwaitinfo]
trace_back: #2 0xfc155594 [event_signal_thread]

```

ENDOFSTACKTRACE

```

.
.
.

```

```

-----
Output from top_procs
Computing times...Unable to enter cbreak mode.: Inappropriate I/O control operat
ion
Error entering control break mode

```

```

node0_RP0_CPU0: 97 procs, 1 cpu, 1.04 delta, 00:03:56 uptime
Memory: 4096 MB total, 3.630 GB free, sample time: Wed Dec 21 17:47:41 2005
cpu 0 idle: 93.22%, kernel: 0.29%

```

pid	mem MB	user	cpu	kernel	cpu	delta	% ker	% tot	name
28691	0.371	0.108	0.003	0.064	0.00	6.37	devb-ata		
405603	0.109	0.008	0.010	0.002	0.19	0.19	top_procs		
41001	0.531	0.087	0.061	0.001	0.09	0.09	dsc		
45103	2.132	0.166	0.098	0.001	0.00	0.09	wdsysmon		
28694	36.304	0.230	0.188	0.000	0.00	0.00	eth_server		
86101	0.792	0.111	0.034	0.000	0.00	0.00	shelfmgr		
77896	1.054	0.175	0.063	0.000	0.00	0.00	gsp		
32792	0.484	0.100	0.012	0.000	0.00	0.00	bcm_process		
32794	0.097	0.020	0.011	0.000	0.00	0.00	stp_process		

```

32802      0.234      0.353      0.120      0.000      0.00      0.00 sysmgr

Output from top_procs
-----
Exiting at  at 17:47:41.487 UTC Wed Dec 21 2005.

```

## Shutdown Collection Scripts

Shutdown scripts are generated when a failure occurs and contains information on why the failure occurred. The system attempts to gather as much information as possible upon failure.

```
RP/0/RP0/CPU0:router# dir harddisk:/shutdown
```

```
Directory of harddisk:/shutdown
```

```

3683      drwx  4096      Wed Dec 21 17:49:34 2005  node0_RP0_CPU0
3672      drwx  4096      Sun Mar 12 15:48:20 2006  node0_1_CPU0
241041792 -rwx  12334      Wed Dec 28 16:00:36 2005  node0_1_CPU0.log.first.gz
241041888 -rwx  11181      Sun Mar 12 15:48:08 2006  node0_1_CPU0.log.next.gz

```

```
39929724928 bytes total (39908044800 bytes free).
```

The shutdown scripts are saved as a zipped file (.gz). The following example shows how to view the zipped files using Ksh commands.

```

RP/0/RP0/CPU0:router# run
# gzip -d node0_1_CPU0.log.first.gz
# cat node0_1_CPU0.log.first.gz

```

## ASIC error Collection Scripts

ASICs are monitored for errors. There are preset thresholds for each ASIC which allows a specific number of errors before an action is taken.

ASIC error scripts are generated when major thresholds are exceeded and the node is reloaded. The scripts contain a snapshot of error types and number of errors for the ASIC on which they occurred.

```
RP/0/RP0/CPU0:router# dir harddisk:/ASIC-ERROR
```

```
Directory of harddisk:/ASIC-ERROR
```

```

1881      drwx  4096      Thu Jun 16 08:32:14 2005  node0_3_CPU0
2141      drwx  4096      Fri Oct 14 13:52:22 2005  node0_5_CPU0

```

See the “ASIC Errors” section on [page 1-49](#) for information on ASIC errors.

## Monitoring

Monitoring allows you to view interface, controller fabric, or controller SONET counters, and auto-updating statistics on processes and threads in real-time. The following commands are used for monitoring:

- [monitor interface Command, page 1-67](#)
- [monitor controller Command, page 1-68](#)

- [monitor processes Command, page 1-69](#)
- [monitor threads Command, page 1-70](#)

## monitor interface Command

Use the **monitor interface** command to monitor interface counters. The following example shows the output of the **monitor interface** command.



### Note

The Cisco IOS XR Manageability Package is required to use the **monitor interface** command.

The following example shows the output of the **monitor interface** command with a specified interface.

```
RP/0/RP0/CPU0:router# monitor interface tenGigE 0/3/0/7
```

```
CRS-A_IOX          Monitor Time: 00:00:08          SysUptime: 118:58:20
```

```
TenGigE0/3/0/7 is up, line protocol is up
Encapsulation ARPA
```

```
Traffic Stats:(2 second rates)                                Delta
Input  Packets:                1466462                        4
Input  pps:                    1
Input  Bytes:                  379782697                      1730
Input  Kbps (rate):            6                             ( 0%)
Output Packets:                2403444                        2
Output pps:                    0
Output Bytes:                  269350468                      140
Output Kbps (rate):            0                             ( 0%)
```

```
Errors Stats:
Input  Total:                  2                             0
Input  CRC:                    0                             0
Input  Frame:                  0                             0
Input  Overrun:                0                             0
Output Total:                  0                             0
Output Underrun:               0                             0
```

```
Quit='q', Freeze='f', Thaw='t', Clear='c', Interface='i',
Next='n', Prev='p'
```

```
Brief='b', Detail='d', Protocol(IPv4/IPv6)='r'
```

The following example shows the output of the **monitor interface** command.

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

```
CRS-A_IOX          Monitor Time: 00:00:08          SysUptime: 118:58:50
```

Interface	In(bps)	Out(bps)	InBytes/Delta	OutBytes/Delta
MgmtEth0/RP0/CPU0/0	45014/ 0%	52156/ 0%	3.9G/14686	842.2M/17016
Bundle-POS100	0/ --%	0/ --%	0/0	0/0
Bundle-POS101	0/ --%	0/ --%	0/0	0/0
MgmtEth0/RP1/CPU0/0	0/ 0%	0/ 0%	27.8M/0	1.8M/0
FINT0/RP1/CPU0	0/ 0%	0/ 0%	121.8M/0	3.5M/0
TenGigE0/3/0/0	0/ 0%	0/ 0%	0/0	0/0
TenGigE0/3/0/1	443/ 0%	683/ 0%	532.1M/144	744.4M/222
TenGigE0/3/0/2	0/ 0%	0/ 0%	0/0	0/0
TenGigE0/3/0/3	0/ 0%	0/ 0%	0/0	0/0

```

TenGigE0/3/0/4          0/ 0%          0/ 0%          0/0          0/0
TenGigE0/3/0/5          0/ 0%        5342/ 0%        2.4M/0        166.8M/1743
TenGigE0/3/0/6          0/ 0%          0/ 0%        11.5T/0        11.5T/0
TenGigE0/3/0/7          852/ 0%        963/ 0%       379.7M/276     269.3M/312
TenGigE0/3/0/6.10       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/6.20       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/6.30       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/6.40       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/6.50       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/6.60       0/ 0%          0/ 0%        1.8T/0        1.8T/0
TenGigE0/3/0/1.99       0/ 0%          0/ 0%       5040/0        4978/0
POS0/5/0/0              0/ 0%          0/ 0%          0/0          0/0
POS0/5/0/1             14658/ 0%       796/ 0%      611.8M/4764   657.5M/259
POS0/5/0/2              0/ 0%          0/ 0%          0/0          2.2M/0
POS0/5/0/3              0/ 0%          0/ 0%          0/0          70.0M/0
POS0/5/0/4             20346/ 0%      20033/ 0%    992.7M/6638   990.7M/6536
POS0/5/0/5              306/ 0%        306/ 0%        4.0M/100      4.0M/100
POS0/5/0/6              0/ 0%          0/ 0%          0/0          0/0
POS0/5/0/8              0/ 0%          0/ 0%          0/0          2.2M/0
POS0/5/0/10             729/ 0%        649/ 0%       97.8M/238     121.5M/212
POS0/5/0/11             723/ 0%      14187/ 0%     99.5M/238     648.7M/4664
POS0/5/0/12             0/ 0%          0/ 0%          0/0          0/0

```

```

Quit='q',      Clear='c',      Freeze='f', Thaw='t',
Next set='n', Prev set='p', Bytes='y', Packets='k'

```

## monitor controller Command

Use the **monitor controller** command to monitor controller fabric or SONET counters in real-time. The counters are refreshed every two seconds. The following example shows the output of the **monitor controller** command.

```
RP/0/RP0/CPU0:router# monitor controller sonet 0/1/0/0
```

```

CRS-8_X1          Monitor Time: 00:00:00          SysUptime: 19:02:19

CRS-8_X1          Monitor Time: 00:00:02          SysUptime: 19:02:21

CRS-8_X1          Monitor Time: 00:00:04          SysUptime: 19:02:23

CRS-8_X1          Monitor Time: 00:00:06          SysUptime: 19:02:25

CRS-8_X1          Monitor Time: 00:00:08          SysUptime: 19:02:27

CRS-8_X1          Monitor Time: 00:00:10          SysUptime: 19:02:29

CRS-8_X1          Monitor Time: 00:00:12          SysUptime: 19:02:31

CRS-8_X1          Monitor Time: 00:00:14          SysUptime: 19:02:33

CRS-8_X1          Monitor Time: 00:00:16          SysUptime: 19:02:35

```

```

CRS-8_X1                Monitor Time: 00:00:18                SysUptime: 19:02:38

CRS-8_X1                Monitor Time: 00:00:20                SysUptime: 19:02:40

Controller for SONET0/1/0/0                0 ( 0 per-se                Delt
  Path LOP                0 ( 0 per-sec)                0
Controller Stats:                0 ( 0 per-se                Delt
  Path LOP                0 ( 0 per-sec)                0
  Path AIS                0 ( 0 per-sec)                0
  Path RDI                0 ( 0 per-sec)                0
  Path BIP                0 ( 0 per-sec)                0
  Path FEBE                0 ( 0 per-sec)                0
  Path NEWPTR                0 ( 0 per-sec)                0
  Path PSE                0 ( 0 per-sec)                0
  Path NSE                0 ( 0 per-sec)                0
  Line AIS                0 ( 0 per-sec)                0
  Line RDI                0 ( 0 per-sec)                0
  Line BIP                0 ( 0 per-sec)                0
  Line FEBE                0 ( 0 per-sec)                0
  Section LOS                1 ( 0 per-sec)                0
  Section LOF                0 ( 0 per-sec)                0
  Section BIP                0 ( 0 per-sec)                0

Quit='q', Freeze='f', Thaw='t', Clear='c'

```

The output allows you to verify if the SONET interfaces have any path errors (Layer 2). Information about the operational status of SONET layers on a particular SONET port is displayed. The output is the same as the **show controllers sonet** command except the display refreshes every 2 seconds.

## monitor processes Command

Use the **monitor processes** command to display the top ten processes based on CPU usage in real time. The display refreshes every 10 seconds.

```

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

Computing times...
235 processes; 822 threads; 4468 channels, 5805 fds
CPU states: 98.0% idle, 0.3% user, 1.5% kernel
Memory: 4096M total, 3492M avail, page size 4K

      JID TIDS Chans   FDs Tmrs   MEM   HH:MM:SS   CPU   NAME
str
      1   28   238    15    1     0   15:50:01   1.58%  procnto-600-smp-cisco-in
      57    5   238   833    0     4M   0:00:10   0.13%  dllmgr
      75   12   230     9    3     1M   0:02:05   0.03%  qnet
     145    4    29    39    5   408K   0:00:01   0.03%  devc-vty
      53    1    1     7    0   108K   0:00:01   0.03%  bcm_logger
      52    5   15     9    4   708K   0:00:07   0.01%  bcm_process
     249    3   52    37    9     1M   0:00:00   0.01%  lpts_pa
     109    5    5    13    3   756K   0:00:00   0.01%  bcdl_agent
    65554    7   16     3    3     7M   0:02:33   0.01%  devb-ata
     291   19   22    67    5   828K   0:00:01   0.01%  raw_ip

```

See [Chapter 8, “Process Monitoring and Troubleshooting,”](#) for more information on processes and process monitoring.

## monitor threads Command

Use the **monitor threads** command to display the top ten threads based on CPU usage in real time. The display refreshes every 10 seconds.

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

```
Computing times...
```

```
235 processes; 822 threads;
CPU states: 96.7% idle, 0.9% user, 2.2% kernel
Memory: 4096M total, 3492M avail, page size 4K
```

JID	TID	PRI	STATE	HH:MM:SS	CPU	COMMAND
1	25	10	Run	0:00:16	2.24%	procnto-600-smp-cisco-instr
65754	1	10	Rply	0:00:00	0.53%	top
59	7	55	Rcv	0:01:41	0.04%	eth_server
59	1	10	Rcv	0:00:47	0.04%	eth_server
308	5	10	Rcv	0:00:21	0.04%	shelfmgr
59	3	50	Sem	0:00:40	0.04%	eth_server
308	1	10	Rcv	0:00:15	0.04%	shelfmgr
341	18	10	Rcv	0:00:02	0.04%	udp
261	9	10	Rcv	0:00:17	0.04%	netio
261	4	10	Rcv	0:00:10	0.04%	netio

## Gathering Information for Technical Support

The following commands are useful for gathering information about your system before you contact Cisco Technical Support. We recommend that you run these commands if time permits, especially if your system has experienced a major problem.

- show install active summary
- show version
- show platform
- show run
- show context location all
- show log
- show inventory
- show diag
- show processes cpu
- show ip interface brief
- show route vrf-route all summary

In addition, you should upload any coredumps that were written to disk0:, disk1: or harddisk:.

The following commands can also be useful in revealing problems:

- show arm conflicts
- show arm ipv6 conflicts
- show placement



- show install log reverse
- show tech-support
- admin show dsc
- admin show sdr summary
- admin show redundancy summary
- admin show redundancy
- admin show controllers fabric plane all detail

Before contacting Cisco Technical Support, review the information provided at the following URL:  
<http://www.cisco.com/web/services/ts/access/index.html>.

For more information on gathering system information, see [Chapter 7, “Collecting System Information”](#).





## CHAPTER 2

# Troubleshooting Booting

---

This chapter describes techniques that you can use to troubleshoot a router running Cisco IOS XR software. It includes the following sections:

- [Booting Tips, page 2-73](#)
- [Verifying Successful Bootup, page 2-74](#)
- [Verifying and Troubleshooting the Console Connection, page 2-74](#)
- [Verifying and Troubleshooting Route Processor and Shelf Controller Cards, page 2-75](#)

## Booting Tips

The following booting sequence should be followed. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for detailed information on the booting sequence for routers running Cisco IOS XR software.

1. Connect to the console port.
2. Boot the route processor (RP) of the designated shelf controller (DSC).  
For Cisco CRS-1 Multishelf Systems, when the RP of the DSC boots up, if the RP or shelf controller (SC) on other chassis are configured to autoboot (for example, they are in a loop sending minimum boot image [MBI] requests), they boot automatically. If they have not been configured to autoboot, enter **reset** at the ROM Monitor (ROMMON) prompt.
3. Verify that the value of the configuration register (confreg) is 0x102. Enter **confreg** at the ROMMON prompt to display the configuration register value. If the configuration register value is not set to 0x102, see *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on changing the configuration register value.
4. It takes approximately 20 minutes for the cards on all chassis to be displayed in the Cisco IOS XR software **show** commands. Managed nodes (service processor [SP], line card [LC], and distributed route processor [DRP]) are booted by shelfmgr, mbimgr, and instdir automatically when ROMMON is set to autoboot.

Use the following chassis numbering guidelines when numbering the chassis in a Cisco CRS-1 Multishelf System:



### Note

It is recommended that the line card chassis containing the active DSC be numbered 0, the second line card chassis be numbered 1, and the fabric card chassis be numbered F0. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for more information on recommended chassis numbering.

- The line card chassis (LCC) numbering must be in the range of 0 to 127.
- The fabric card chassis (FCC) numbering must be in the range of 240 to 255.
- In a multishelf configuration, all racks must have a unique number.

## Verifying Successful Bootup

For information on verifying the status of the router, see *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router*. It contains detailed procedures for verifying a standalone router and multishelf system.

The **show platform** command displays the state transitions when a node is booting:

- SP RMON—The service processor (SP) is starting and the basic code is being loaded on the SP (LC only).
- ROMMON—Once the SP has started, the LC CPU is on and the LC ROMMON software is started (LC only).
- MBI BOOT—The minimum boot image code is loading on the LC CPU from disk or the DSC (LC only).
- MBI RUNNING—The required processes on the LC are starting. The normal transition from this state is to start the Cisco IOS XR operating system.
- IOS XR—The Cisco IOS XR software is running.
- PRESENT—LC is not sending boot request or boot request is dropped

## Verifying and Troubleshooting the Console Connection

The default configuration selects the route processor (RP) in the lowest numbered slot. For the Cisco CRS-1 Multishelf System, the RP in the lowest numbered slot in the LCC configured as Rack 0 is selected as the primary RP and the designated shelf controller (DSC).

To determine the primary RP on routers running Cisco IOS XR software:

- On Cisco CRS-1 RPs, the primary RP is identified by a lighted Primary LED on the RP front panel. The primary RP is also identified by “ACTV RP” in the alphanumeric display on the card. The console port is located on the primary RP.

The console connection through the DSC provides a communications path to the routing system. The console port is designed for a serial cable connection to a terminal or computer running a terminal emulation program. See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for information on connecting to the console port.

To verify and troubleshoot the console connection on the DSC, perform the following procedures.

### SUMMARY STEPS

1. Check the cable between the console port on the DSC and the terminal or computer.
2. Open the terminal emulation application, select the correct COM port and verify the settings.
3. Swap cables.
4. Test the cable on another console port.
5. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	Check the cable between the console port on the DSC and the terminal or computer.	Ensure that there is a serial cable connection from a terminal or computer running a terminal emulation program to the console port on the active RP of the DSC. See <i>Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router</i> for information on connecting to the console port.
<b>Step 2</b>	Open the terminal emulation application, select the correct COM port and verify the settings.	<p>Verify that the terminal settings are as follows:</p> <ul style="list-style-type: none"> <li>• Bits per second: 9600/9600</li> <li>• Data bits: 8</li> <li>• Parity: None</li> <li>• Stop bit: 2</li> <li>• Flow control: None</li> </ul> <p>If the correct settings are not applied, the console port outputs garbage characters or no output at all. This makes the router appear as if it is hanging or not responding.</p> <p>If this fails to solve the problem, proceed to <a href="#">Step 3</a>.</p>
<b>Step 3</b>	Swap cables.	Swap cables.
<b>Step 4</b>	Test the cable on another console port.	Test the console cable by attaching the console cable to the console port of another router.
<b>Step 5</b>	Contact Cisco Technical Support.	If these steps do not resolve the problem, contact Cisco Technical Support. See the “ <a href="#">Obtaining Documentation and Submitting a Service Request</a> ” section on page viii in the <a href="#">Preface</a> .

## Verifying and Troubleshooting Route Processor and Shelf Controller Cards

The following RP and SC verification and troubleshooting procedures are provided:

- [Troubleshooting RP and SC Cards Not Booting, page 2-75](#)
- [Troubleshooting RP and SC Cards Resetting While Booting, page 2-78](#)
- [Troubleshooting Blocked FCC Shelf Controller or LCC Route Process Minimum Boot Image Requests, page 2-79](#)

### Troubleshooting RP and SC Cards Not Booting

To troubleshoot the RP and SC cards in a router running Cisco IOS XR software, perform the following procedure.

Troubleshooting procedures for the following cards is provided in this section:

- Cisco CRS-1 4-Slot Line Card Chassis/Cisco CRS-1 8-Slot Line Card Chassis/Cisco CRS-1 16-Slot Line Card Chassis
  - Primary (DSC) and standby route processors (RPs)
- Cisco CRS-1 Multishelf System
  - DSC in the line card chassis (LCC)
  - Standby shelf controller (SC) in the LCC
  - SC in the fabric card chassis (FCC)
  - Primary and standby route processor (RP) in the LCC

## SUMMARY STEPS

1. Place the RP (or SC) in ROMMON mode.
2. **set**
3. Reseat the RP (or SCs) in the affected chassis.
4. Swap RP slots.
5. Verify connection status.
6. Collect console messages generated by the RP.
7. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	Place the RP or SC not booting in ROMMON mode.	See Router Recovery with ROM Monitor in <i>Cisco IOS XR ROM Monitor Guide</i> for information on entering ROMMON mode.
Step 2	<b>set</b>  <b>Example:</b> rommon B1 > set	Displays the environment variable settings for the card. Verify that the variables are valid. The following are valid variables: <pre>rommon B1 &gt; set PS1=rommon ! &gt; ?=0</pre> The following variables are the main variables used for netboot: <pre>IP_SUBNET_MASK=255.255.0.0 TFTP_SERVER=223.255.254.254 IP_ADDRESS=12.2.53.41 TFTP_FILE=muck/jasamson/comp-hfr-full-hfr34-qq.vml DEFAULT_GATEWAY=12.2.0.1</pre> The following variables add verbosity to certain ROMMON functions and to bypass auxiliary authentication in Cisco IOS XR: <pre>TFTP_VERBOSE=2 AUX_AUTHEN_LEVEL=0</pre>

Command or Action	Purpose
	<p>The following variables are used by the Config Manager to immediately apply the configuration after bootup:</p> <pre>IOX_ADMIN_CONFIG_FILE=nvram:startup-config-admin.00 IOX_CONFIG_FILE=nvram:startup-config-lr.00</pre> <p>The following variables above may be specified may be displayed as follows if a path has not been specified:</p> <pre>IOX_CONFIG_FILE= IOX_ADMIN_CONFIG_FILE=</pre> <p>The following variable may be displayed before the system has been Turboboooted:</p> <pre>TURBOBOOT=on, format, disk0</pre> <p>The following variables may be displayed after a system has been Turboboooted. A 'boot' variable is saved to indicate where the minimum boot image (MBI) is stored and the turboboot variable is set to NULL:</p> <pre>BOOT=disk0:hfr-os-mbi-3.3.96/mbihfr-rp.vm,1 TURBOBOOT=</pre> <p>The following variable is Reboot/Reload code saved after any kind of shutdown, failover, and so on:</p> <pre>ReloadReason=0</pre> <p>The following variable is saved to NVRAM.</p> <pre>BSI=0</pre>
<b>Step 3</b> Remove then reinsert the SC or RP.	<p>See the following documents for information on removing then reinserting the SC (or RP):</p> <ul style="list-style-type: none"> <li>• <i>Installing the Cisco CRS-1 Carrier Routing System 16-Slot Line Card Chassis</i></li> <li>• <i>Installing the Cisco CRS-1 Carrier Routing System 8-Slot Line Card Chassis</i></li> </ul> <p>The documents are located at the following URL:</p> <p><a href="http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html">http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html</a></p>
<b>Step 4</b> Swap RP (or SC) slots.	<p>Swap the RP (or SC) card in the chassis:</p> <ul style="list-style-type: none"> <li>• RP slots—RP0 and RP1</li> <li>• SC slots—SC0 and SC1</li> </ul>

	Command or Action	Purpose
Step 5	Verify connection status.	<p>For the Cisco CRS-1 Multishelf System:</p> <ul style="list-style-type: none"> <li>Use the <b>show controllers switch</b> <i>switch-instance statistics location 0/RP0/CPU0</i> command in administration mode to verify the switch statistics. Repeat this command for switches on each shelf in the system.</li> <li>For the SC on the FCC and the primary and standby RP on the LCC, verify the system control plane Ethernet network connection status between the DSC and card that will not boot. See the <a href="#">“Troubleshooting the Multishelf System Router Topology”</a> section on page 6-153 of Chapter 6, <a href="#">“Troubleshooting the Control Plane Ethernet Network,”</a> for information on verifying connection status between the DSC and the SC or RP.</li> </ul>
Step 6	Collect console messages generated by the RP (or SC).	See <a href="#">Chapter 7, “Collecting System Information”</a> for details on the information required when contacting Cisco Technical Support.
Step 7	Contact Cisco Technical Support.	For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Troubleshooting RP and SC Cards Resetting While Booting

To troubleshoot resetting RP and SC cards on Cisco CRS-1 systems, perform the following procedures.

### SUMMARY STEPS

1. Place both RPs in ROMMON mode.
2. **set**
3. Verify that all three LEDs on power modules are green.
4. Power cycle the power modules.
5. Check that the DSC is active.
6. Collect console messages generated by the RP.
7. Contact Cisco Technical Support if the problem is not resolved.



## DETAILED STEPS

	Command or Action	Purpose
Step 1	Place both RPs in ROMMON mode.	See Router Recovery with ROM Monitor mode. See the <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for information on entering ROMMON mode.
Step 2	<b>set</b>  <b>Example:</b> rommon B1 > set	Displays the environment variable settings for the card. Verify that the variables are valid.
Step 3	For each affected chassis, verify that all three LEDs on power modules are green.	See <i>Cisco CRS-1 Carrier Routing System 16-Slot Line Card Chassis Hardware Operations and Troubleshooting Guide</i> for the location of the power modules.
Step 4	Power cycle the power modules to confirm that the LEDs transition from amber to green.	If the LEDs on the power modules stay amber or go from amber to green then back to amber, there is a problem with the power module.  See the following documents for power cycling information: <ul style="list-style-type: none"> <li>• <i>Installing the Cisco CRS-1 Carrier Routing System Fabric Card Chassis</i></li> <li>• <i>Installing the Cisco CRS-1 Carrier Routing System Line Card Chassis</i></li> </ul> The documents can be found at the following URL: <a href="http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html">http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html</a>
Step 5	Check that the DSC is active.	Verify that the Primary LED on the DSC front panel is lit.
Step 6	Collect console messages generated by the RP.	See <a href="#">Chapter 7, “Collecting System Information”</a> for details on the information required when contacting Cisco Technical Support.
Step 7	Contact Cisco Technical Support.	See the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the <a href="#">Preface</a> for Cisco Technical Support contact information.

## Troubleshooting Blocked FCC Shelf Controller or LCC Route Process Minimum Boot Image Requests

**Note**

This procedure applies to Cisco CRS-1 Multishelf Systems only.

To troubleshoot blocked fabric card chassis SC or line card chassis RP minimum boot image (MBI) requests, perform the following procedures.

If the shelf controller (SC) or route processor (RP) in the remote line card chassis fails to boot correctly, messages are displayed on the console of either device, indicating that it is waiting for information from the designated shelf controller (DSC).

## SUMMARY STEPS

1. Verify the physical system control plane Ethernet network connection between the DSC in the LCC to the Catalyst switch.
2. Connect to the Catalyst switch console port.
3. Verify the physical system control plane Ethernet network connection between the SC in the FCC to the Catalyst switch.
4. **show spanning-tree**
5. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	Verify the physical system control plane Ethernet network connection between the DSC in the LCC and the Catalyst switch.	Verify the physical connectivity between the DSC and Catalyst switch. See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on cabling between the DSC and the Catalyst switch.
Step 2	Connect to the Catalyst switch console port.	Provides connection to the system control plane Ethernet network.  See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on connecting to the external switch.
Step 3	Verify the system control plane Ethernet network.	Verify the system control plane Ethernet network connection status between the DSC and SC. See the <a href="#">“Troubleshooting the Multishelf System Router Topology”</a> section on page 6-153 in Chapter 6, <a href="#">“Troubleshooting the Control Plane Ethernet Network”</a> for information on verifying the system control plane Ethernet network connection status between the DSC and SC.
Step 4	<b>show spanning tree</b>  <b>Example:</b> Router# show spanning-tree	Displays the spanning tree port states.  Verify that the ports used to connect the DSC, remote LCC RP, and the FCC SC are in the forwarding state. The listed interfaces should include the port to which you have connected.  If the port is not listed, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.
Step 5	Contact Cisco Technical Support.	See the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface for Cisco Technical Support contact information.



## CHAPTER 3

# Troubleshooting Forwarding

---

Cisco Express Forwarding (CEF) is the mechanism that enables packet forwarding. CEF information is examined when data forwarding is not occurring as expected. Troubleshooting CEF involves comparing the Routing Information Base (RIB) information to the software Forwarding Information Base (FIB), verifying that the hardware is programmed correctly, verifying that the adjacencies are built correctly, verifying the control plane is built correctly, and gathering any necessary trace information.

The only prerequisite for CEF is a valid route in the RIB.

This chapter describes techniques that you can use to troubleshoot router forwarding. It includes the following sections:

- [Troubleshooting IPv4 CEF Information, page 3-81](#)
- [Troubleshooting Adjacency Information, page 3-85](#)
- [Troubleshooting Transient Traffic Drop in Forwarding, page 3-88](#)
- [Troubleshooting Control Plane Information, page 3-90](#)
- [Troubleshooting the Interface Manager, page 3-95](#)
- [Troubleshooting the Interface Manager Distributor, page 3-106](#)

## Troubleshooting IPv4 CEF Information

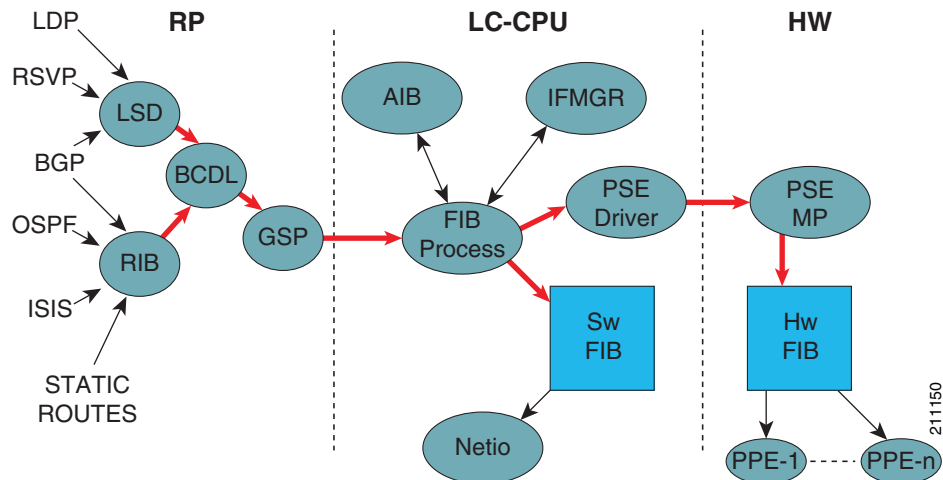
CEF is an advanced, Layer 3 IP switching technology that optimizes network performance. It also improves the scalability for networks with large and dynamic traffic patterns, such as the Internet and networks characterized by intensive Web-based applications.

Information conventionally stored in a route cache is stored in several data structures for CEF switching. The data structures provide optimized lookup for efficient packet forwarding. The two main components of CEF operation are forwarding information base (FIB) and adjacency tables:

- CEF uses a FIB to make IP destination prefix-based switching decisions. FIB maintains a mirror image of the forwarding information contained in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next hop address information based on the information in the IP routing table. There is a one-to-one correlation between FIB entries and routing table entries, therefore FIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.
- Nodes in the network are said to be adjacent if they can reach each other with a single hop across a link layer. In addition to the FIB, CEF uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Figure 3-1 provides an overview of the components involved in contributing information to the CEF process and displays the interaction between the software and hardware elements of the CEF process.

**Figure 3-1 CEF Process**



To troubleshoot IPv4 CEF information on Cisco IOS XR software, perform the following procedure.

## SUMMARY STEPS

1. **show route ipv4 prefix mask**
2. **show cef ipv4 prefix mask detail**
3. **show cef ipv4 prefix mask detail location node-id** (on ingress line card)
4. **show cef ipv4 prefix mask detail location node-id** (on egress line card)
5. **show cef ipv4 prefix mask hardware ingress detail location node-id**
6. **show cef ipv4 prefix mask hardware egress detail location node-id**
7. **show cef ipv4 interface type instance location node-id**
8. **show cef ipv4 summary location node-id**
9. **show cef ipv4 trace location node-id**
10. **show cef platform trace ipv4 all location node-id**
11. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>show route ipv4 prefix mask</b></p> <p><b>Example:</b> RP/0/RP0/CPU0:router# show route 192.168.2.0 255.255.255.0</p>	<p>Displays the current routes in the Routing Information Base (RIB).</p> <ul style="list-style-type: none"> <li>Check the prefix and mask, as well as the next hop and outgoing interface, to ensure that they are what is expected.</li> <li>Note the timer value that shows how long the route has been in the routing table. If the timer value is low the route may be flapping.</li> </ul> <p>A lower timer value is present when a route is installed in the RIB for a short period of time. A low timer value may indicate flapping. For example, if a BGP route was being installed and removed from the RIB table every sixty seconds, then the route is flapping.</p> <p>Look for routes that have not been installed in the routing table for very long. The route will either be stable or flapping. If the route is flapping, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.</p> <ul style="list-style-type: none"> <li>Check that route is learned via the routing protocol you are expecting it to be known via, and that the metric is what you expect.</li> </ul>
Step 2	<p><b>show cef ipv4 prefix mask detail</b></p> <p><b>Example:</b> RP/0/RP0/CPU0:router# show route ipv4 192.168.2.0 255.255.255.0 detail</p>	<p>Displays the IPv4 Cisco Express Forwarding (CEF) table detailed entry information.</p> <ul style="list-style-type: none"> <li>Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the <b>show route ipv4 prefix mask</b> command.</li> <li>Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node.</li> <li>Check that the expected hash (load balance) and egress interfaces are listed.</li> </ul>

	Command or Action	Purpose
Step 3	<p><b>show cef ipv4 prefix mask detail location node-id</b></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 detail location 0/14/cpu0</p>	<p>Displays the IPv4 CEF table for the designated ingress node.</p> <ul style="list-style-type: none"> <li>Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the <b>show route ipv4 prefix mask</b> command.</li> <li>Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node.</li> <li>Check that the expected hash (load balance) and egress interfaces are listed.</li> </ul>
Step 4	<p><b>show cef ipv4 prefix mask detail location node-id</b></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 detail location 0/13/cpu0</p>	<p>Displays the IPv4 CEF table for the designated egress node.</p> <ul style="list-style-type: none"> <li>Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the <b>show route ipv4 prefix mask</b> command.</li> <li>Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node.</li> <li>Check that the expected hash (load balance) and egress interfaces are listed.</li> </ul>
Step 5	<p><b>show cef ipv4 prefix mask hardware ingress detail location node-id</b></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 hardware ingress detail location 0/14/cpu0</p>	<p>Displays the IPv4 CEF table for the designated ingress node.</p> <ul style="list-style-type: none"> <li>Check that the prefix and mask are valid.</li> <li>Check the nexthop IP address is as expected</li> <li>Check that the entry type is set to forward.</li> <li>Check that the hardware and software representations in hex format match. For example:</li> </ul> <pre>SW: 0x0c000000 00000020 00000000 00000000 HW: 0x0c000000 00000020 00000000 00000000</pre>
Step 6	<p><b>show cef ipv4 prefix mask hardware egress detail location node-id</b></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 hardware detail egress location 0/13/cpu0</p>	<p>Displays the IPv4 CEF table for the designated egress node.</p> <ul style="list-style-type: none"> <li>Check that the prefix and mask are valid.</li> <li>Check the nexthop IP address is as expected</li> <li>Check that the entry type is set to forward.</li> <li>Check that the hardware and software representations in hex format match. For example:</li> </ul> <pre>SW: 0x0c000000 00000020 00000000 00000000 HW: 0x0c000000 00000020 00000000 00000000</pre>

	Command or Action	Purpose
Step 7	<b>show cef ipv4 interface</b> <i>type instance location node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef ipv4 interface tengige 1/3/0/7 location 1/3/cpu0	Displays IPv4 CEF-related information for an interface.  Verify the interface handle ‘interface is marked’ is as expected. The command output also shows how many references there are to the interface in CEF table and the IPv4 MTU.  Use this command for the ingress and egress interfaces.
Step 8	<b>show cef ipv4 summary location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef ipv4 summary location 0/3/cpu0	Displays a summary of the IPv4 CEF table. Check the VPN routing and forwarding (VRF) names associated with the node, the route update drops, and that there are the expected number of incomplete adjacencies.  Note the number of routes CEF has entries for, the number of load sharing elements, and the number of references to this node.  Use this command for the ingress and egress line cards and route processor (RP).
Step 9	<b>show cef ipv4 trace location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef ipv4 trace location 0/3/cpu0	Displays IPv4 CEF trace table information.  Check if there is any flap on the prefix.  Use this command for the RP, and ingress and egress interfaces for the local line card.
Step 10	<b>show cef platform trace ipv4 all location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef platform trace ipv4 all location 0/3/cpu0	Displays CEF IPv4 hardware status and configuration trace table information.  Verify that the TLU pointer and RPF pointer are as expected.  Use this command for the ingress and egress interfaces for the local line card.
Step 11	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Troubleshooting Adjacency Information

To troubleshoot adjacency information on Cisco IOS XR software, perform the following procedure.

### SUMMARY STEPS

1. **show arp location** *node-id*
2. **show arp traffic location** *node-id*
3. **show adjacency interface-type interface-instance remote detail location** *node-id*
4. **show adjacency interface-type interface-instance remote detail hardware location** *node-id*
5. **show adjacency ipv4 nexthop ipv4-address detail location** *node-id*
6. **show adjacency interface-type interface-instance detail location** *node-id*

7. **show adjacency ipv4 nexthop** *ipv4-address detail hardware location node-id*
8. **show adjacency interface-type interface-instance detail hardware location node-id**
9. **show adjacency trace location node-id**
10. **show adjacency trace client aib-client location node-id**
11. **show adjacency hardware trace location node-id**
12. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show arp location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show arp location 0/12/cpu0	Displays the Address Resolution Protocol (ARP) for an egress line card with a broadcast interface.  Ensure that you can find the IP address and that correct MAC address of the neighbor is learned.
Step 2	<b>show arp traffic location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show arp traffic location 0/12/cpu0	Displays ARP traffic statistics for an egress line card with a broadcast interface.  Check for any errors or IP packet drops.
Step 3	<b>show adjacency interface-type interface-instance remote detail location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency pos 0/13/0/2 remote detail location 0/14/cpu0	Displays detailed CEF adjacency table information for a remote ingress line card.  Ensure that the output shows IPv4 adjacency information and that an adjacency exists.
Step 4	<b>show adjacency interface-type interface-instance remote detail hardware location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency pos 0/13/0/2 remote detail hardware location 0/14/cpu0	Displays adjacency information for a remote ingress line card. <ul style="list-style-type: none"> <li>• Check that the prefix and mask are valid.</li> <li>• Check that the table look-up (TLU) pointers match the TLU pointers in the <b>show cef ipv4 prefix mask hardware ingress detail location node-id</b> command. For example:</li> </ul>
Step 5	<b>show adjacency ipv4 nexthop</b> <i>ipv4-address detail location node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency ipv4 nexthop 192.168.2.0 detail location 0/12/cpu0	Displays adjacencies on an egress line card with a broadcast interface that are destined to the specified IPv4 next hop.  When an egress interface is broadcast, use the <b>show adjacency ipv4 nexthop</b> command to display the adjacency information.  Compare the mac layer rewrite information that shows the destination L2 address in the first part followed by the source L2 address, and the Ethernet value with the output from the <b>show arp location node-id</b> command.



	Command or Action	Purpose
Step 6	<b>show adjacency</b> <i>interface-type</i> <i>interface-instance</i> <b>detail location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency pos 0/13/0/2 detail location 0/13/cpu0	Displays CEF adjacency table information for an egress line card with a point to point interface.  There should be two IPv4 entries in the command output. Ensure both entries exist. <ul style="list-style-type: none"> <li>• The SRC MAC only entry is used for multicast switching</li> <li>• The point to point entry is used for unicast switching.</li> </ul> On broadcast interfaces you will have a SRC MAC only and one for each nexthop IP address. Please note the MTU is for the IPv4 minus the layer 2 header. Use the <b>show im chains</b> command to display MTU details.
Step 7	<b>show adjacency ipv4 nexthop</b> <i>ipv4-address</i> <b>detail hardware location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency ipv4 nexthop 192.168.2.0 detail hardware location 0/12/cpu0	Displays the hardware programming associated with the adjacency. Verify that the packets are being switched in the hardware.
Step 8	<b>show adjacency</b> <i>interface-type</i> <i>interface-instance</i> <b>detail hardware location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency pos 0/13/0/2 detail hardware location 0/13/cpu0	Displays the hardware programming information for a point-to-point interface such as the Packet-over-SONET/SDH (POS) interface. The rewrite information is slightly different because there is no MAC rewrite string as there is in Ethernet.  Verify that the rewrite is appropriate for the encapsulation on the interface. Compare the CEF hardware output and verify that the pointer matches the egress adjacency.
Step 9	<b>show adjacency trace location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency trace location 0/13/cpu0	Displays CEF adjacency trace table information.  Use this command for the egress interfaces for the local line card.
Step 10	<b>show adjacency trace client</b> <i>aib-client</i> <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency trace client ipv4_fib_mgr location 0/13/cpu0	Displays CEF adjacency trace table information for a specified adjacency information base (AIB) client.  Use this command for the egress interfaces for the local line card.
Step 11	<b>show adjacency hardware trace location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show adjacency hardware trace location 0/13/cpu0	Displays CEF adjacency hardware trace table information.  Use this command for the egress interfaces for the local line card.
Step 12	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The following example shows that the TLU pointers match. The TLU pointers are indicated in bold.

```
RP/0/RP0/CPU0:router# show cef 202.112.36.224/30 hardware ingress location 0/0/CPU0

202.112.36.224/30, version 1536, internal 0x42040001[1]
.
.
.
TLU2 0x0101041d
    TLU2 ENTRY          0
    SW: 0x00000000 00000000 00000000 00a22800
    HW: 0x00000000 00000000 00000000 00a22800
    label1:          0    label2:          0
    num of labels:    0    next ptr: 0x0000a228

RP/0/RP0/CPU0:router# show adjacency pos 0/4/0/15 remote detail hardware location 0/0/CPU0
.
.
.
ingress adjacency
    TLU3                : 0x200a228
    [HW: 0x00400000 0x00000000 0x00000000 0x00082800]
        num. entries    : 1
        num. labels      : 0
        label 1          : 0
        label 2          : 0
        next ptr         : 0x828
    TLU4                : 0x3000828
    Entry[0]
        [HW: 0x00000000 0x11410000 0x01480600
```

The following example shows that the address information matches. The addresses are indicated in bold.

```
RP/0/RP0/CPU0:router# show arp location 0/1/cpu0

Address          Age          Hardware Addr   State   Type   Interface
212.27.50.157    02:08:34    0016.c761.f509 Dynamic  ARPA   TenGigE0/1/0/2

RP/0/RP0/CPU0:router# show adjacency ipv4 nexthop 212.27.50.157 detail loccation 0/1/cpu0

Interface          Address                      Version  Refcount  Protocol
TenGigE0/1/0/2     212.27.50.157                41        2         ipv4
0016c761f5090015fa9959890800
mtu: 1500, flags 0 0 0
2894 packets, 156876 bytes
0xffffffff
```

# Troubleshooting Transient Traffic Drop in Forwarding

To troubleshoot a momentary drop in IPV4 packet forwarding on Cisco CRS-1 router, perform the following procedure.

## SUMMARY STEPS

1. `show interfaces interface name`
2. `show controller <ingressq/egressq> qmstats 0 location <line card>`

3. **show controller pse** [ingress|egress] **statistics location** *node-id*
4. **show captured packets** [ingress|egress] **location** *node_id*
5. **show route** <prefix> **det**
6. **show cef** <prefix> **det location** *location-id*
7. **show cef** <prefix> **hardware ingress detail location** *location-id*
8. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show interfaces</b> <i>interface name</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show interface POS0/5/0/0	Displays the packet count in the ingress and egress interfaces before and after the transmission of packets.  If the packets are dropped either in ingress or egress interface, then stop here and contact the Cisco Technical Support team for troubleshooting the plim interfaces.
Step 2	<b>show controller</b> <ingressq egressq> <b>qmstats 0 location</b> <i>location-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controller ingressq qmstats 0 location 0/12/cpu0	Checks the statistics in buffer management ASIC (BMA) at both ingress and egress line cards before and after transmission.  If you see any drop count incrementing other than soft drop, then contact the Cisco Technical Support team for resolution. If there is no increment in the drop count or only the soft drop is incrementing, continue with the further steps.
Step 3	<b>show controller pse</b> <ingress egress> <b>statistics location</b> <i>location-id</i>  <b>Example:</b> RP/0/9/CPU0:router#show controller pse ingress statistics loc 0/5/cpu0   inc drop RP/0/9/CPU0:EE10-1#show controller pse ingress statistics loc 0/5/cpu0   inc mtu	Displays the packets that are dropped at the packet switching engine (PSE), the drop count, and maximum transmission unit (mtu) count.  Ensure that the show command is executed multiple times in order to check if the count is actually incrementing. To check if the route has been removed and re-installed, you must identify the traffic stream or prefix which experienced the drop first.
Step 4	<b>show captured packets</b> [ingress egress] <b>location</b> <i>node_id</i>	Displays the packets which are dropped by the PSEs on the line card. The next step is to find out what packets get dropped. The debug command displays the L2 and L3 information of packets which are dropped by the PSE microcode. When the captured packet is an MPLS packet, you must look for the label of the packet. The label will be used as the key to verify the forward chain. When the captured packet is an IPv4 or IPv6 packet, you must look for the destination address of the packet. The address is used as the key to verify the forward chain of the prefix.
Step 5	<b>show route</b> <prefix> <b>det</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show route 20.20.1.2	Checks the routing table and the timestamp provides you information about when the route was installed.

	Command or Action	Purpose
Step 6	<pre>show cef &lt;prefix&gt; det show cef &lt;prefix&gt; det location <i>location-id</i></pre> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show cef ipv4 nexthop  192.168.2.0 detail location 0/12/cpu0</p>	Checks if the route is installed in FIB or CEF is platform-independent. If the output of this show command indicates that the CEF is pointing to drop adjacency, then you can stop here and contact the Cisco Technical Support team.
Step 7	<pre>show cef &lt;prefix&gt; hardware ingress detail location <i>location-id</i></pre>	Checks if the Cisco Express Forwarding information is embedded in the hardware. If the MTU in the output is 5 or 6, then it indicates that the route has been programmed as drop in hardware.

## Troubleshooting Control Plane Information

To troubleshoot control plane information on Cisco IOS XR software, perform the following procedure.

### SUMMARY STEPS

1. **show netio idb** *interface-type interface-instance location node-id*
2. **show uidb index**
3. **show uidb data** *interface-type interface-instance location node-id*
4. **show im chains** *interface-type interface-instance location node-id*
5. **show imds interface brief**
6. **show tbn hardware {ipv4 | ipv6} unicast dual detail location node-id**
7. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>show netio idb</b> <i>interface-type interface-instance location node-id</i></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show netio idb tengige0/0/0/0 location 0/0/cpu0</p>	<p>Displays control plane information for the software switching path. The output provides useful statistics for determining software forwarding issues.</p> <ul style="list-style-type: none"> <li>• Verify the encap and decap paths</li> <li>• Ensure that all of the appropriate steps in the chain are shown for all of the features that may be enabled on the interface.</li> </ul> <p><b>Note</b> Fixup is a direct pointer to a routine in the output path after a CEF rewrite. this is an optimized path if a CEF rewrite exists and can be used.</p> <ul style="list-style-type: none"> <li>• Verify that the ifhandle and global uidb value is correct.</li> </ul> <p>Use this command for the ingress and egress interfaces for the local line card.</p>
Step 2	<p><b>show uidb index</b></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show uidb index</p>	<p>Displays the micro-interface descriptor block (IDB) index assigned by the software.</p> <p>Check that the interface and the universal interface descriptor block (UIDB) value are what is expected.</p> <p>Compare the IDB index to the uidb index value in the <b>show adjacency ipv4 interface-type interface-instance detail hardware location node-id</b> command output.</p>
Step 3	<p><b>show uidb data</b> <i>interface-type interface-instance location node-id</i></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router# show uidb data tengige 1/3/0/0 location 0/3/cpu0</p>	<p>Displays, from a software perspective, features that are enabled on a selected interface.</p> <ul style="list-style-type: none"> <li>• Check the UIDB value.</li> <li>• Check what flags are enabled for the UIDB.</li> <li>• Check the ifhandle in the UIDB to make sure it is correct.</li> </ul> <p>Compare the output to the configuration of the interface and expected features.</p> <p>Use this command for the ingress and egress interfaces for the local line card.</p>

	Command or Action	Purpose
<b>Step 4</b>	<b>show im chains</b> <i>interface-type</i> <i>interface-instance</i> <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show im chains pos 0/14/0/0 location 0/14/cpu0 or RP/0/RP0/CPU0:router# show im chains pos 0/13/0/2 location 0/13/cpu0	Displays the output of the control plane encapsulations for data plane forwarding. <ul style="list-style-type: none"> <li>Check the different layers for the interface, the status (up or down) of each layer, and the maximum transmission unit (MTU) at each layer.</li> <li>Verify the ifhandle value the ingress line card will use to forward packets that are destined out of the interface.</li> </ul> Compare the output to the expected encapsulations on the interface, the correct MTU values, and the correct ifhandle value from the <b>show cef ipv4 interface</b> command output.  Use this command for the ingress interface on the ingress line card and the egress interface on egress line card.
<b>Step 5</b>	<b>show imds interface brief</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show imds interface brief	Displays interface manager distribution server (IMDS) interface information.  <b>Note</b> This is just a partial output not full output.  Check the state, MTU, encapsulation being used, and the ifhandle for each interface.
<b>Step 6</b>	<b>show tbm hardware {ipv4   ipv6} unicast dual detail</b> <i>location</i> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show tbm hardware ipv4 unicast dual detail location 0/13/cpu0 or RP/0/RP0/CPU0:router# show tbm hardware ipv4 unicast dual detail location 0/14/cpu0	Displays tree bitmap hardware-related ingress and egress information.  Check if there have been any failures in the different stages of the lookup.  Use this command for the ingress and egress interfaces for the local line card.
<b>Step 7</b>	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The following example displays the control plane information for the software switching path. Check for any errors or drops.

```
RP/0/RP0/CPU0:router# show netio idb tenGigE 0/1/1/0 location 0/1/cpu0
```

```
TenGigE0/1/1/0 (handle: 0x01180020, nodeid:0x11) netio idb:
```

```
-----
name:                               TenGigE0_1_1_0
interface handle:                    0x01180020
interface global index:              2
physical media type:                 30
dchain ptr:                          <0x482ae8e0>
echain ptr:                          <0x482d791c>
fchain ptr:                          <0x482d79b8>
driver cookie:                       <0x4824ad58>
driver func:                         <0x4824ad44>
number of subinterfaces:             4096
```

```

subblock array size:      3
DSNCF:                   0x00000000
interface stats info:
  IN  unknown proto pkts: 0
  IN  unknown proto bytes: 0
  IN  multicast pkts:     0
  OUT multicast pkts:     0
  IN  broadcast pkts:     0
  OUT broadcast pkts:     0
  IN  drop pkts:          0
  OUT drop pkts:          0
  IN  errors pkts:        0
  OUT errors pkts:        0

Chains
-----
Base decap chain:
  ether                <30>  <0xfd7aef88, 0x48302824>  <      0,      0>

Protocol chains:
-----
<Protocol number> (name) Stats
  Type Chain_node      <caps num> <function, context> <drop pkts, drop bytes>
<7> (arp)      Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
  Encap:
    l2_adj_rewrite    <86>  <0xfcec7a88, 0x4834efec>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>
  Decap:
    queue_fifo        <56>  <0xfcedda4c, 0x482dbee4>  <      0,      0>
    arp               <24>  <0xfd1082cc, 0x00000000>  <      0,      0>
  Fixup:
    l2_adj_rewrite    <86>  <0xfcec745c, 0x00000000>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>
<12> (ipv4)     Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
  Encap:
    ipv4              <26>  <0xfd10f41c, 0x482d7724>  <      0,      0>
    ether             <30>  <0xfd7aeb44, 0x48302824>  <      0,      0>
    l2_adj_rewrite    <86>  <0xfcec7a88, 0x4834f104>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>
  Decap:
    queue_fifo        <56>  <0xfcedda4c, 0x482dbee4>  <      0,      0>
    ipv4              <26>  <0xfd10f474, 0x00000000>  <      0,      0>
  Fixup:
    l2_adj_rewrite    <86>  <0xfcec745c, 0x00000000>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>
<22> (ether_sock) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
  Encap:
    ether_sock        <98>  <0xfd7b1630, 0x48302824>  <      0,      0>
    l2_adj_rewrite    <86>  <0xfcec7a88, 0x48304c1c>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>
  Decap:
    queue_fifo        <56>  <0xfcedda4c, 0x482dbee4>  <      0,      0>
    ether_sock        <98>  <0xfd7b1874, 0x48302824>  <      0,      0>
  Fixup:
    l2_adj_rewrite    <86>  <0xfcec745c, 0x00000000>  <      0,      0>
    queue_fifo        <56>  <0xfcedda68, 0x482dbee4>  <      0,      0>
    txm_nopull        <60>  <0xfcea2a5c, 0x482dc11c>  <      0,      0>

```

Protocol SAFI counts:

Protocol	SAFI	Pkts In	Bytes In	Pkts Out	Bytes Out
ipv4	Unicast	0	0	0	0
ipv4	Multicast	0	0	0	0
ipv4	Broadcast	0	0	0	0
ipv6	Unicast	0	0	0	0
ipv6	Multicast	0	0	0	0

The following example shows that the micro-idb index value is 12.

```
RP/0/RP0/CPU0:router# show uidb index tengige1/3/0/6.30 location 1/3/cpu0
```

Location	Interface-name	Interface-Type	Ingress-index	Egress-index
1/3/CPU0	TenGigE1_3_0_6.30	Sub-interface	20	12

Comparing the IDB index value of 12 in the **show uidb index** command to the uidb index value in the following command output shows that the values are the same.

```
RP/0/RP0/CPU0:router# show adjacency ipv4 tengige1/3/0/6.30 detail hardware location 1/3/cpu0
```

```
Interface                      Address                      Version  Refcount  Protocol
TenGigE1/3/0/6.30             (src mac only)              90       1         ipv4
                                000000000000001243602d8b8100001e0800
                                mtu: 1500, flags 1 0 1
                                453 packets, 42582 bytes
                                453 hw-only-packets, 42582 hw-only-bytes

ether egress adjacency
TLU1                           : 0x4407
[HW: 0x00401862 0xc4170800 0x8100001e 0x01060700]
  num. entries   : 1
  uidb index     : 12
  counter msb    : 0x2
  counter lsb    : 0xc417
  vlan e or len  : 0x800
  ether len      : 0x8100 (33024)
  vlan info      : 30
  next ptr       : 0x10607
```

The following example displays, from a software perspective, features that are enabled on a selected interface. Compare the output to the configuration of the interface and expected features. Verify that the configured features are correctly enabled.

```
RP/0/RP0/CPU0:router# show uidb data location 0/6/cpu0
```

```
-----
Location = 0/6/CPU0
Index = 0
Pse direction = INGRESS

Global general 16 bytes:
-----
ROUTER_ID: 45.104.151.108
MINIMUM MASK DESTINATION: 0 / 0
MINIMUM MASK SOURCE: 0 / 0
BYTES OF SNIFF PACKET: 0
SUPPRESS PUNT ACL: 0
MPLS PROPAGATE TTL FLAG: 1
PARITY: 0
```



```

FABRIC QOS ENABLE FLAG: 0
-----
Location = 0/6/CPU0
Index = 0
Pse direction = EGRESS

Global general 16 bytes:
-----
ROUTER_ID: 45.104.151.108
MINIMUM MASK DESTINATION: 0 / 0
MINIMUM MASK SOURCE: 0 / 0
BYTES OF SNIFF PACKET: 0
SUPPRESS PUNT ACL: 0
MPLS PROPAGATE TTL FLAG: 1
PARITY: 0
IPV4 PREFIX ACCNTG: 0
-----

Location = 0/6/CPU0
Ifname/Ifhandle = GigabitEthernet0_6_5_0
Index = 1
Pse direction = INGRESS

General 16 bytes:
-----
IFHANDLE: 0x168002
STATUS: 0
IPV4 ENABLE: 0
IPV6 ENABLE: 0
MPLS ENABLE: 0
STATS POINTER: 0x2c400
SPRAYER QUEUE: 32
IPV4 MULTICAST: 0
IPV6 MULTICAST: 0
USE TABLE ID IPV4: 0
USE TABLE ID IPV6: 0
USE TABLE ID MPLS: 0
TABLE ID: 0
QOS ENABLE: 0
QOS ID: 0
NETFLOW SAMPLING PERIOD: 0
L2 PKT DROP: 0
L2 QOS ENABLE: 0
SRC FWDING: 0
*BUNDLE IFHANDLE: 0
*TUNNEL IFHANDLE: 0
*L2 ENCAP: 3

* Not programmed in hardware
.
.
.

```

## Troubleshooting the Interface Manager

These sections describe how to troubleshoot the interface manager (IM):

- [Interface Manager Control Process, page 3-96](#)
- [Troubleshooting the Trace Logs for the Interface Manager, page 3-100](#)
- [Troubleshooting the Client for the Interface Manager, page 3-101](#)

- [Troubleshooting the Rules for the Interface Manager, page 3-101](#)
- [Troubleshooting the Control Chain and Interface Information, page 3-103](#)
- [Troubleshooting the Registrations for the Interface Manager, page 3-105](#)

## Interface Manager Control Process

The interface manager (IM) is the main control process in the Packet Forwarding Infrastructure (PFI) and runs one copy on each line card (LC), route processor (RP), and distributed route processor (DRP). The IM process creates interfaces, adds protocols and encapsulations, initiates state and MTU walks, and registers for notification changes.

The following sample output from the **show im server activity** command with the **summary** keyword shows an overall summary of information about the instance of IM on a particular node:

```
RP/0/RP0/CPU0:router# show im server activity summary
```

```
INTERFACE MANAGER ACTIVITY SUMMARY (v1.0)
```

### Notifications

```
-----
Owner                                     :    2298
Third Party                             :    2058
Total notifications sent                 :    4356
-----
```

### Control tree nodes

```
-----
              intf:      caps: [      ctrl:]      proto:      vc:
-----
Created              506      1037 [      1543]      511      3
Deleted                0         0 [         0]         6      0
Existing              506      1037 [      1543]      505      3
-----
Memory (bytes):      194304      157624 [      351928]      67682      49212
-----
      Total memory (bytes):      468822
Avg. memory per intf (bytes):      926
      Avg. caps per intf:      2
      Increases of intf proto map:      1
      Increases in proto max:      2
-----
```

### Client requests

```
-----
Request Type              Requests:  Operations:      Max Ops:
-----
SPECIFY_CONTROL              23           23           1
NOTIFY_GET                  136          136           1
INTERFACE_REPLICATE          0            0            0
INTERFACE_CREATE             4            4            1
INTF_CREATE_WITH_BCAPS       0            0            0
INTERFACE_DELETE             0            0            0
DELETE_CONFIRM               0            0            0
CAPS_BASE_DEFINE             0            0            0
RESOURCE_ALLOC_DONE          0            0            0
CAPS_ADD                     6            6            1
CAPS_REMOVE                  0            0            0
ADOPT_CONFIRM                0            0            0
PROTO_LOOKUP                 5            5            1
PROTO_REVERSE_LOOKUP         0            0            0
CAPS_LOOKUP                  10           10            1
-----
```

CAPS_REVERSE_LOOKUP	1	1	1
INTF_TYPE_LOOKUP	3	3	1
INTF_TYPE_REVERSE_LOOKUP	0	0	0
INTERFACE_FIND	29	29	1
INTERFACE_NAME	223	223	1
INTERFACE_GET_BASE_CAPS	7	7	1
INTERFACE_READY	4	4	1
INTERFACE_GET_TYPE	3	3	1
CONTROL_PARENT_INTF_LOOKUP	0	0	0
INTERFACE_DESC_LOOKUP	0	0	0
CAPS_DISPLAY_NAME_LOOKUP	0	0	0
STATE_QUERY	3263	3263	1
STATE_REGISTER	1	1	1
STATE_UPDATE	4	4	1
CREATE_REGISTER	9	9	1
CREATE_REGISTER_NAME	0	0	0
CHILDREN_CREATE_REGISTER	0	0	0
OWNER_SUPPORT_SET	1	1	1
OWNER_SUPPORT_GET	0	0	0
OWNER_CHANGE	0	0	0
CONFIG_NEW_ENCAP	0	0	0
IF_CONFIGURABLE	0	0	0
IF_IS_HW	0	0	0
MTU_UPDATE	4	4	1
MTU_IMPOSE	0	0	0
MTU_QUERY	1	1	1
MTU_REGISTER	1	1	1
DATA_CAPS_LOAD	0	0	0
DATA_CAPS_UNLOAD	0	0	0
IF_IS_CONTROL_ONLY	0	0	0
GET_MAIN_INT_PARSER_DATA	0	0	0
GET_SUB_INT_PARSER_DATA	0	0	0
REMOVE_CALLBACK	0	0	0
REPLACE_BASE_CAPS	0	0	0
CAPS_DPC_DISTRIBUTE	5	5	1
PROTO_CHAINS_DISABLE	0	0	0
PROTO_CHAINS_ENABLE	0	0	0
-----			
IF_CREATE	3	502	500
IF_READY	0	0	0
IF_FIND	578	1578	500
IF_DELETE	0	0	0
DEL_CONFIRM	0	0	0
IF_UPDATE_PRIMARY	1	1	1
IF_FLAGS	0	0	0
CAPS_ADD	26	526	500
CAPS_REMOVE	0	0	0
CAPS_BASE_DEFINE	3	3	1
ID_LOOKUP	9	9	1
STATE_UPDATE	14	14	1
MTU_UPDATE	17	202	93
REG_WALK	9	11	2
REG_CREATE	27	29	3
IF_NAME	13	14	2
STATE_QUERY	0	0	0
MTU_QUERY	0	0	0
PIC_UPDATE	0	0	0
RSRC_ALLOC	3	3	1
REPL_INTF_OPER	0	0	0
REPL_INTF_OPER_ACK	0	0	0
INTF_CONFIG_PARAM	0	0	0
IF_REPLICATE	0	0	0
ADOPT_CONFIRM	0	0	0
CFG_NEW_BC	0	0	0

REPLACE_BC	0	0	0
IF_TYPE_GET	3	3	1
IF_DESC_LOOKUP	0	0	0
IF_DESC_LOOKUP_BYNAME	0	0	0
CAPS_DN_LOOKUP	0	0	0
CTRLP_IF_LOOKUP	0	0	0
ID_REV_LOOKUP	0	0	0
REG_CREATE_NAME	0	0	0
IF_GET_BC	1	1	1
DPC	19	521	500
DP_RESTART_REG	2	3	2
IF_CREATE_CHILDREN	0	0	0
IF_CREATE_IND	0	0	0
CAPS_ADD_SP	0	0	0
CAPS_ADD_TGT	0	0	0
CAPS_GET_NODESET	0	0	0
IIR_UL_INFO	0	0	0
CFG_RESTORE	0	0	0
Total non-bulk requests:	3743	3743	1
Total bulk requests:	728	3420	500

Request data received: 230058 bytes

IM to IMProxy download commands

	#cmds	#bytes	#mallocs
Total	5891	135328	11782
Per cmd	-	22	2

Downloads

#Super	#GS_SHM	Total [	Empty	Errors	Timeouts]
3	55	58 [	1	0	0]

Total data downloaded: 169652 bytes

Avg. size of download: 2925 bytes

Download Memory (in bytes)

Type	Reserved	In Use	Max In Use	Max Dnld
Super	-	0	106032	106032
GS_SHM	821688	0	12196	12196
Atomic Ring	448016	400000	-	-

Rules

Max Protocol:	:	85
Max Protocol Chg:	:	2

Interface Handle Allocation Memory: (element is 16 bytes)

Allocation Type: Initial Sz: Chunk Sz: #Chunks: Max Chunks:

Physical interface handles	0	32	16	16
Virtual interface handles	0	32	0	0
Other physical handles	0	512	4	4
Other virtual handles	0	512	0	0

Initial Memory: 0 bytes

Current Memory: 40960 bytes

Max Memory: 40960 bytes

Interface Handle Freelist Memory (element is 16 bytes)

Freelist Type	Size	Max Size	Memory
Physical interface handles	518	544	8288
Virtual interface handles	0	0	0
Other physical handles	2551	2560	40816
Other virtual handles	0	0	0
Current usage	3069		49104
Max usage		3104	49664

PFI-IFH Broadcast Queue Memory (element is 20 bytes)

Allocation Type: Initial Sz: Chunk Sz: #Chunks: Max Chunks:

Create Queue	1024	1024	0	0
Delete Queue	0	0	0	0

Current Total Memory: 20480 bytes

Max Memory: 20480 bytes

1500 items committed in 23 sec (62)items/sec

Updating

Timer Queue (element is 24 bytes)

Queue	Size	Max Size	Memory
State	11	11	264
MTU	706	706	16944
Delete	0	0	0
Current Total	717	717	17208
Max Memory			17208
Freelist	0	8	0

Bulk stores

Store	Entries	Max Entries	Memory
Bulk	0	0	0
Bulk Delete	0	0	0

Updated Commit database in 1 sec

ulk Delete	0	0	0
DP Restart	2	2	56
Total	2	2	56

Client callback information

#Active	#Dead	Total	Memory (bytes)
23	0	23	19568

Requested Client Connections	:	23
Requested Disconnections	:	0
Unrequested Disconnections	:	0
IM server (re)start count	:	1

```

Third Party Registrations
Create/Delete          State          MTU      Memory (bytes)
-----
36                     6              7         1072
-----

IM->IMD communications information
Type      Chunk Sz    #Chunks    Used      Max      Mem      Max Mem
-----
Big        1024         30         0          0         0         0
Small      512          300        44        1358       880       106800
No-chunk    0            0          0          0         0         0
Total                      330        44        1358       880       106800
-----

Pre-allocated memory for small chunks      :    155057 bytes
Pre-allocated memory for big chunks        :     31095 bytes
-----

Transmit queue size: 142 elements
[range MIN:100 -> MAX:1000
 based on nfn size: 108 bytes, optimal msg size: 15376 bytes]
-----

Comms type    Count    #Elem  Average  Success  #Retry    Fail    #Retry
Tx start      1875    945489    504      72        0        0        0
Tx done        73     4054      55
-----

Total bytes sent:      186332
      attempted:      186332
-----

Checkpoint information
-----

Timer Expiries          :          234
Checkpoints             :           19
-----

Node status      Chkptd Avg nodes/chkpt    Bytes chkptd Avg bytes/chkpt
Clean            300          15        35360          1861
Dirty            2616         137       353984         18630
Total            2916         153       389344         20491
-----

Memory summary
-----

Total memory reserved      :    1251438 bytes
Total memory used          :     997670 bytes
-----

```

## Troubleshooting the Trace Logs for the Interface Manager

To debug the problem that occurred on the router and the IM, use the **show im trace** command with both the **location** and **internal** keywords to trace logs that are recovered from the router for the applicable location.

The **debug im errors** command enables the server for error debugging. When the problem is narrowed down, debugging is enabled to provide more information.

The **debug im callbacks** command is used to display when notifications are arriving at the clients from IM or Interface Manager Distributor (IMD). In addition, the command checks to see the number and types of messages that are in the set of notifications.

## Troubleshooting the Client for the Interface Manager

A client must be connected to use any of the IM services. If the client is an owner of the nodes in the IM or is registered for the notifications with the IM, the client must specify a callback handle. The **show im client** command is used to find which clients are connected and have registered callback handles.



### Note

The *context* is a unique parameter that the IM uses to identify a client and the registration. The IM is allowed to know when a client has reconnected to the IM after a process restart. Because the IM tracks the client's status, the **show im client** command is also used to display which clients have previously connected to the IM. If there is no connection, the status is dead.

The following sample output from the **show im client** command shows which clients are connected:

```
RP/0/RP0/CPU0:router# show im client info
```

```
INTERFACE MANAGER ACTIVITY CLIENT-INFO
```

Process	PID	JID	Context	Status	Handle
pfi_ifh_server	53321	246	pfi_ifh	alive	0x30000001
sysldr	53297	287	slr_im_evc	alive	0x30000002
driver_infra_partner	24603	54	di-node0_0_CPU0-2	alive	0x30000003
ether_caps_partner	61515	154	ether	alive	0x30000004
qos_ma	61517	254	qos_ma	alive	0x30000005
driver_infra_partner	24603	54	caps_netio_0	alive	0x30000006
aib	61520	100	aib	alive	0x30000007
ipv4_fib_mgr	61525	188	ipv4_fib_cfg_im_connect	alive	0x30000008
arp	61522	102	arp	alive	0x30000009
arp	61522	102	arp_ipv4_caps_reg	alive	0x3000000a
ipv4_io	65623	189	ipv4	alive	0x3000000b
ntpd	65632	192	ntpv4	alive	0x3000000c
cdp	65633	125	cdp	alive	0x3000000d
cdp	65633	125	cdp_caps	alive	0x3000000e
mpls_lfd	65631	227	mpls_lfd	alive	0x3000000f
tcp	65640	294	tcp	alive	0x30000010
clns	65643	130	clns_transport	alive	0x30000011
parser_server	65657	244	parser	alive	0x30000012
bundlemgr_distrib	65653	124	BM-DISTRIB_LOCAL	alive	0x30000013
nd_partner	65671	235	Null0	alive	0x30000014
ipv6_io	65670	202	ipv6-fint	alive	0x30000015
null_caps_partner	69770	239	null_caps_partner	alive	0x30000016
ipv6_nd	65672	204	ipv6-nd	alive	0x30000017
ipv6_grp	69785	201	ipv6-grp	alive	0x30000018
mpls_lsd	69788	229	mpls_lsd	alive	0x30000019
fint_partner	53302	157	FINT	alive	0x30000000

## Troubleshooting the Rules for the Interface Manager

A process that calls the IM to create an interface or add a capsulation is defined as an *owner*. The owners own sets of triplets from an interface, protocol, or capsulation. The triplets are secure domain router (SDR) unique identifiers to control the nodes.

The interface handle is displayed in hexadecimal and contains platform-specific mapping for the node ID, virtual bit, and interface instance number for the node ID.

The protocol and capsulation numbers are small numbers and are displayed in hexadecimal or decimal. They represent a unique identifier for a specific protocol such as IPv4, IPv6, or MPLS.

The **show im rules** command is used to set interfaces, protocols, and capsulations that are installed on the router.

The following sample output from the **show im rules** command with the **interface-types** keyword shows the types of interfaces:

```
RP/0/RP0/CPU0:router# show im rules interface-types
```

```
IM rules interface-type data
=====
tag:                IFT_LOOPBACK
id:                 16
source:             /pkg/rules/loopback.intf
description:        Loopback interface(s)
allowed-base-caps:  loopback
default-base-caps:  loopback
dynamic-ifname:     Loopback
owner-string:       loopback
default-mtu:        1514
subint-range:       0 - 65535
invisibility:       0x2
ignore:             mtu delete
```

The following sample output from the **show im rules** command with the **capsulations** keyword shows the rule of each type:

```
RP/0/RP0/CPU0:router# show im rules capsulations
```

```
IM rules caps data
=====
tag:                ipv6_preroute
id:                 128
source:             /pkg/rules/ipv6.caps
description:        IPv6 preroute capsulation nodes
dll:                libipv6_netio.dll
ordering:           0x6fffffff
switching:          decaps encaps
ignore:             state mtu
```

The following sample output from the **show im rules** command with the **protocols** keyword shows the rules for the protocols:

```
RP/0/RP0/CPU0:router# show im rules protocols
```

```
IM rules proto data
=====
tag:                fint_n2n
id:                 6
source:             /pkg/rules/fint_n2n.caps
description:        Forwarder netio 2 netio packet forwarding protocol
dll:                libfint_n2n.dll
chain:              use-base-caps
```

If the rules for a specific capsulation, protocol, or interface are missing from the IM on a given node, you cannot create a control node for the type. If the rules are on the router, a process restart of ifmgr can resynchronize the rules.



## Troubleshooting the Control Chain and Interface Information

When the control nodes are created, you can use both the **show im chains** command and the **show im children** command to see the given location and state for the IM nodes. Both commands display the ASCII names, which map to the interface, protocol, and capsulation triplets.



**Tip**

Keep the following tips in mind when you are troubleshooting:

- Only configurable interfaces display the names that are in the list. If you use the **location**, **all**, and **ifhandle** keywords, the system displays all relevant interfaces (for example, SONET).
- Take care when using the **location** keyword. If you specify the wrong location, the queried IM does not contain the chains that you are looking for. For example, the POS 0/2/0/0 location does not display anything.
- The **show im chains** command is also used as an interface handle (ifhandle) to find a name.

The following sample output from the **show im chains** command shows all the interface PICs and the current interface flags:

```
RP/0/RP0/CPU0:router# show im chains location all
```

```
Showing all interface control chains:
```

```
-----
Interface MgmtEth0/0/CPU0/0, ifh 0x01000100 (up)
[pic:0x0, intf_flags:0x5]
Protocol      Caps (state, mtu)
<base>        txm_nopull (up, 1514)
               queue_fifo (up, 1514)
               ether (up, 1514)
arp           arp (up, 1500)
ipv4          ipv4 (up, 1500)
ether_sock    ether_sock (up, 1500)

Interface Null0, ifh 0x01000080 (up)
[pic:0x0, intf_flags:0x1c]
Protocol      Caps (state, mtu)
<base>        null (up, 1500)

...

Interface POS0/2/0/0, ifh 0x03000400 (administratively down)
[pic:0x0, intf_flags:0x15]
Parent interface: SonetPath0_2_0_0, ifh 0x03000300
Protocol      Caps (state, mtu)
<base>        txm_nopull (administratively down, 4474)
               queue_fifo (administratively down, 4474)
               hdlc (administratively down, 4474)
chdlc         chdlc (administratively down, 4470)
               slarp (administratively down, 4470)

...

Controller SONET0/3/0/0, ifh 0x04000200 (administratively down)
[pic:0x0, intf_flags:0x37]
```

The following sample output from the **show im children** command with the **ifhandle** keyword shows when the subinterfaces or other control parent and child summaries are required:

```
RP/0/RP0/CPU0:router# show im children ifhandle 0x03000200
```

```
SONET0/2/0/0, ifh 0x03000200 (administratively down)
  SonetPath0/2/0/0, ifh 0x03000300 (administratively down)
    POS0/2/0/0, ifh 0x03000400 (administratively down)
```

When the interfaces are created, use the **show interfaces** command to examine the state. The following sample output is from the **show interfaces** command for POS 0/2/0/0:

```
RP/0/RP0/CPU0:router# show interfaces POS 0/2/0/0
```

```
POS0/2/0/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: router4 POS 2\0
  Internet address is 3.4.5.6/24
  MTU 4474 bytes, BW 155520 Kbit
    reliability 25/255, txload Unknown, rxload Unknown
  Encapsulation HDLC, crc 16, controller loopback not set, keepalive set (10 sec)
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
    0 runs, 0 giants, 0 throttles, 0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  3 packets output, 62 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets
  0 output errors, 0 underruns, 0 applique, 0 resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

The following sample output from the **show interfaces** command shows the router summary:

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

Interface Type	Total	UP	Down	Admin Down
-----	-----	--	----	-----
ALL TYPES	54	32	1	21
-----				
IFT_ETHERBUNDLE	1	1	0	0
IFT_VLAN_SUBIF	3	3	0	0
IFT_POSBUNDLE	1	1	0	0
IFT_LOOPBACK	1	1	0	0
IFT_NULL	1	1	0	0
IFT_SBC	2	2	0	0
IFT_POS	20	11	1	8
IFT_SERIAL_T3E3	4	0	0	4
IFT_ETHERNET	16	8	0	8
IFT_ETHERNET	4	4	0	0

The following sample output from the **show interfaces** command shows a per interface summary:

```
RP/0/RP0/CPU0:router# show interfaces brief
```

Intf	Intf Name	LineP State	Encap State	MTU	BW Type (byte)	(Kbps)
-----						
	Nu0	up	up		Null 1500	Unknown
	Mg0/0/CPU0/0	up	up		ARPA 1514	100000
	PO0/2/0/0	up	down		HDLC 4474	155520
	PO0/2/0/1	admin-down	admin-down		HDLC 4474	155520
	PO0/2/0/2	admin-down	admin-down		HDLC 4474	155520
	PO0/2/0/3	admin-down	admin-down		HDLC 4474	155520

PO0/7/0/0 admin-down admin-down

HDLC 4474 622080

## Troubleshooting the Registrations for the Interface Manager

When the interfaces and control nodes are created, various processes want to obtain information that includes the notifications of the state, MTU, and existence.

You can use the **show im registrations** command to see which clients are registered with the IM. With the exception of the **all** option, the other options filter the table to show only the rows of a certain type. The **location** keyword is used to direct the command to a specific IM. The job ID (JID) is used with the **show process** command to find more information about the client process, or with the **show im server activity** command with the **client-info** keyword to find out more about the client connection to the IM.

The following sample output from the **show im registrations** command shows all the IM registrations:

```
RP/0/RP0/CPU0:router# show im registrations all
```

### INTERFACE MANAGER REGISTRATIONS

Reg: C - Create, CH - Child create, D - Delete, M - MTU, O - Owner, S - State

Node	JID	Context	Interface Name or Type CH: (parent) type	Protocol	Capsulation	Reg
0	189	ipv4_fib_cfg_f	IFT_FINT_INTF	0	0	C
0	244	pfi_ifh	any	0	any	C
0	102	arp	IFT_ETHERNET	unknown	unknown	C
0	100	aib	any	ipv6	ipv6	C
0	100	aib	any	0	0	C
0	102	arp	IFT_ETHERNET	unknown	unknown	C
0	100	aib	FINT0_0_CPU0	ipv6	ipv6	M
0	102	arp_ipv4_caps_	MgmtEth0_0_CPU0_0	ipv4	ipv4	S
48	148	ipv4_fib_cfg_f	IFT_FINT_INTF	0	0	C
48	102	arp	unknown	unknown	unknown	C
48	173	pfi_ifh	any	0	any	C
48	102	arp	IFT_ETHERNET	unknown	unknown	C
48	101	aib	any	ipv6	ipv6	C
48	101	aib	any	0	0	C
48	102	arp	IFT_ETHERNET	unknown	unknown	C
48	50	SONET-0_3_CPU0	(SonetPath0_3_0_3)	any	0	CH
48	50	POS- 30	POS0_3_0_3	0	0	S
48	50	SONET-0_3_CPU0	POS0_3_0_3	0	0	S
48	101	aib	FINT0_3_CPU0	ipv6	ipv6	M
112	50	di-node0_7_CPU	Node0_7_CPU0	0	0	O
112	129	ether	GigabitEthernet0_7_0_2	ether_s	ether_sock	O
112	129	ether	GigabitEthernet0_7_0_2	0	ether	O
112	129	ether	GigabitEthernet0_7_0_1	ether_s	ether_sock	O
112	129	ether	GigabitEthernet0_7_0_1	0	ether	O

The registered client receives the changes for the IM notifications. If a client appears to be missing notifications, see the possible reasons in [Table 3-1](#).

**Table 3-1** *List of Reasons for Missed Interface Manager Notifications*

Reason	Solution
Client is not connected to IM.	Check the client connection by using the <b>show im client</b> command with the <b>info</b> keyword.
IM is not running.	Check the interface manager by using the <b>show process</b> command with the <b>ifmgr</b> keyword.
Client is not getting pulses from IM.	Check to see if the client is blocked by using the <b>show process</b> command. Use the <b>debug im callbacks</b> command to debug the callbacks from the IM.
Client is processing the notifications slowly and has notifications queued.	Use the <b>show im client</b> command with the <b>notify</b> keyword to see the activity of the IM.

The following sample output from the **show im client** command with the **notify** keyword shows the IM activity notifications:

```
RP/0/RP0/CPU0:router# show im client notify
```

```
INTERFACE MANAGER ACTIVITY CLIENT-NOTIFY
```

```
-----
```

Context	#Notifications	Max queued
-----	-----	-----
pfi_ifh	7	2
slr_im_evc	1	1
di-node0_0_CPU0-2	4	2

## Troubleshooting the Interface Manager Distributor

These sections provide information on how to troubleshoot the Interface Manager Distributor (IMD):

- [Troubleshooting the Interface Manager Distributor, page 3-106](#)
- [Troubleshooting the Trace Logs for the Interface Manager Distributor, page 3-107](#)
- [Troubleshooting the Clients for the Interface Manager Distributor, page 3-107](#)
- [Troubleshooting the Interface Information for the Interface Manager Distributor, page 3-108](#)
- [Troubleshooting the Global Registrations for the Interface Manager Distributor, page 3-109](#)
- [Troubleshooting the Rules for the Interface Manager Distributor, page 3-110](#)

## Interface Manager Distributor Overview

The IMD is an aggregation service. The following elements are supported:

- Runs one copy on each RP and DRP.
- Discovers information about each control node in each interface manager service on each card through the GSP.
- Contains the interface name, handle, state, MTU, and capsulation state.

- Registers with the IMD for notifications of create, delete, state, and MTU changes for RP and DRP clients (for example, routing protocols) to find information on any interface in the secure domain router (SDR).

IMD cannot be used to update any information.

Every IMD instance is a copy of other instances, because the same updates are received from every IM. Although the IMD runs on the standby cards, the IMD is not aware of the standby issues; instead, the IMD provides a full copy of all information as if it is on an active card and clients can connect, make registrations, and make queries.

## Troubleshooting the Trace Logs for the Interface Manager Distributor

If the problem occurred on the router and affected the IMD, previous information is the most useful. You can use the **show imds trace** command with the **location** and **file** keywords to ensure that the trace logs are recovered from the router for the applicable location.

If there are problems with the connections to the IMD server, use the **debug imd client** command. For problems with the IMD contents (for example, not synchronized with the IM or with the configuration), use the **debug imd imdc** command and **debug imd collector** command. For registrations and notification, use the **debug imd filter** command and the **debug imd notifier** command. For **show** command issues, use the **debug imd edm** command. For IMD update processing, use the **debug imd interface** command and the **debug imd msg** command.

No specific IMD client-side debugging is provided. Because the IMD clients use the IM client library, the same IM client debugging is used.

## Troubleshooting the Clients for the Interface Manager Distributor

To use any of the IMD services, a client must be connected. IMD clients who want to register notifications must specify a callback handle. Part of the function of the application programming interface (API) is to pass a unique client context string.

You can use the **show imds client** command to find out which clients are connected and registered to callback handles, as shown in the following sample output:



### Note

The Filter Ref column refers to the number of registration filters that the client is referencing. If multiple clients make the same set of registrations, the filters are shared between clients. The Interface Ref column refers to the number of interfaces in which the client currently has registrations.

```
RP/0/RP0/CPU0:router# show imds client location 0/0/CPU0
```

```
IMD CLIENTS
```

```
Node: 0x0
```

```
Internal States
```

```
-----
```

```
State: 0x02    LC_req: 0    LC_res: 12    ID: 0x00
Sync: 4        Gather: 0    Resync: 0    Check: 0
```

```
Clients
```

Name	ID	Filter Ref	Interface Ref
-----	-----	-----	-----
ipv4_arm	0x00000064	1	9
ipv6_arm	0x00000065	1	4
arp	0x30000009	4	0

```

parser                0x30000012 2      0
di-node0_0_CPU0-2     0x30000003 0      0
mpls_lsd              0x30000019 0      0
ipv6-grp              0x30000018 1      0
slr_im_evc            0x30000002 1      0
arp_ipv4_caps_reg     0x3000000a 0      1
ISIS_207_BaseCaps     0x00000068 2      0
ISIS_207_CLNS         0x00000069 2      4
ISIS_207_v4/v6        0x0000006a 2      2

```

## Troubleshooting the Interface Information for the Interface Manager Distributor

The **show imds interface** command is used to display the contents of the IMD database. The following sample output from the **show imds interface** command shows the interface information for the IMD database :

```
RP/0/RP0/CPU0:router# show imds interface all location 0/0/CPU0
```

```
IMDS INTERFACE DATA (Node 0x0)
```

```
FINT0_0_CPU0 (0x01000000)
```

```

-----
flags: 0x00000007      type: 27 (IFT_FINT_INTF)      encap: 91 (fint_base)
state: 3 (up)          mtu: 8000      protocol count: 9
control parent: 0x00000000      data parent: 0x00000000

```

protocol	capsulation	state	mtu
0 (Unknown)			
18 (lpts)	91 (fint_base)	3 (up)	6000
6 (fint_n2n)	81 (lpts)	3 (up)	6000
	92 (fint_n2n)	3 (up)	6000
10 (clns)	15 (clns)	3 (up)	6000
12 (ipv4)	26 (ipv4)	2 (down)	0
30 (ipv4_prero)	115 (ipv4_preroute)	3 (up)	6000
13 (mpls)	25 (mpls)	3 (up)	6000
32 (ipv6_prero)	128 (ipv6_preroute)	3 (up)	6000
19 (ipv6)	90 (ipv6_preswitch)	3 (up)	6000
	82 (ipv6)	3 (up)	6000

```
Null0 (0x01000080)
```

```

-----
flags: 0x000000ab      type: 17 (IFT_NULL)      encap: 17 (null)
state: 3 (up)          mtu: 1500      protocol count: 1
control parent: 0x00000000      data parent: 0x00000000

```

protocol	capsulation	state	mtu
0 (Unknown)			
	17 (null)	3 (up)	1500

```
MgmtEth0_0_CPU0_0 (0x01000100)
```

```

-----
flags: 0x0000002f      type: 8 (IFT_ETHERNET)      encap: 30 (ether)

```

```

state: 3 (up)      mtu: 1514      protocol count: 4
control parent: 0x00000000      data parent: 0x00000000
      protocol      capsulation      state      mtu
      -----
0 (Unknown)
      60 (txm_nopull)      3 (up)      1514
      56 (queue_fifo)      3 (up)      1514
      30 (ether)      3 (up)      1514
22 (ether_sock)
      98 (ether_sock)      3 (up)      1500
12 (ipv4)
      26 (ipv4)      3 (up)      1500
7 (arp)
      24 (arp)      3 (up)      1500

SONETO_3_0_0 (0x04000200)
-----
flags: 0x0000006d      type: 22 (IFT_SONET)      encap: 0 (Unknown)
state: 3 (up)      mtu: 10000      protocol count: 0
control parent: 0x00000000      data parent: 0x00000000

SonetPath0_3_0_0 (0x04000300)
-----
flags: 0x00000005      type: 33 (IFT_SONET_PATH)      encap: 0 (Unknown)
state: 3 (up)      mtu: 10000      protocol count: 0
control parent: 0x04000200      data parent: 0x00000000

POS0_3_0_0 (0x04000400)
-----
flags: 0x0000002f      type: 19 (IFT_POS)      encap: 14 (hdlc)
state: 3 (up)      mtu: 4474      protocol count: 4
control parent: 0x04000300      data parent: 0x00000000
      protocol      capsulation      state      mtu
      -----
0 (Unknown)
      60 (txm_nopull)      3 (up)      4474
      56 (queue_fifo)      3 (up)      4474
      14 (hdlc)      2 (down)      4474
9 (chdlc)
      13 (chdlc)      2 (down)      4470
      12 (slarp)      2 (down)      4470
10 (clns)
      15 (clns)      2 (down)      4470
12 (ipv4)
      26 (ipv4)      2 (down)      4470

```

## Troubleshooting the Global Registrations for the Interface Manager Distributor

The **show imds registrations** command is used to display the information for the IMD client registrations.

IMD supports the concept of an encapsulation change registration so that clients are informed of the changes on an interface.

The following sample output from the **show imds registrations** command with the **all** keyword shows all the IMD registrations:

```
RP/0/RP0/CPU0:router# show imds registrations all location 0/0/CPU0
```

```
IMD REGISTRATIONS
```

```
Reg: C - Create, E - Encap, M - MTU, S - State
```

Node	JID	Context	Interface Name or Type	Protocol	Capsulation	Reg
0x0	187	ipv4_arm	any	ipv4	ipv4	C
0x0	198	ipv6_arm	any	ipv6	ipv6	C
0x0	244	parser	any	0	0	C
0x0	201	ipv6-grp	any	ipv6	ipv6	C
0x0	287	slr_im_evc	IFT_OTHER	0	0	C
0x0	102	arp	IFT_ETHERNET	mpls	any	C
0x0	102	arp	IFT_ETHERNET	mpls	any	C
0x0	102	arp	IFT_ETHERBUNDLE	mpls	any	C
0x0	102	arp	IFT_VLAN_SUBIF	mpls	any	C
0x0	207	ISIS_207_BaseC	POS0_3_0_0	0	any	C
0x0	207	ISIS_207_CLNS	0	clns	clns	C
0x0	207	ISIS_207_v4/v6	0	ipv4	ipv4	C
0x0	207	ISIS_207_CLNS	0	clns	clns	C
0x0	207	ISIS_207_v4/v6	0	ipv4	ipv4	C
0x0	244	parser				E
0x0	187	ipv4_arm	FINT0_0_CPU0	ipv4	ipv4	S
0x0	198	ipv6_arm	FINT0_0_CPU0	ipv6	ipv6	S
0x0	187	ipv4_arm	MgmtEth0_0_CPU0_0	ipv4	ipv4	S
0x0	102	arp_ipv4_caps_	MgmtEth0_0_CPU0_0	ipv4	ipv4	S
0x0	187	ipv4_arm	FINT0_3_CPU0	ipv4	ipv4	S
0x0	198	ipv6_arm	FINT0_3_CPU0	ipv6	ipv6	S
0x0	207	ISIS_207_CLNS	POS0_3_0_0	clns	clns	M
0x0	207	ISIS_207_CLNS	POS0_3_0_0	clns	clns	S
0x0	187	ipv4_arm	POS0_3_0_0	ipv4	ipv4	S
0x0	207	ISIS_207_v4/v6	POS0_3_0_0	ipv4	ipv4	S

The following sample output from the **show imds registrations** command shows how the **context** keyword is useful for a particular client callback:

```
RP/0/RP0/CPU0:router# show imds registrations context ipv4_connected all
```

```
IMD REGISTRATIONS
```

```
Context: ipv4_connected
```

Node	JID	Interface Name or Type	Protocol	Capsulation	Reg
0x0	0	any	ipv4	ipv4	Create
0x0	0	any	0	0	Create
0x0	0	MgmtEth0/0/CPU0/0	ipv4	ipv4	State
0x0	0	GigabitEthernet0/7/0/0	ipv4	ipv4	State
0x0	0	GigabitEthernet0/7/0/1	ipv4	ipv4	State
0x0	0	GigabitEthernet0/7/0/2	ipv4	ipv4	State

## Troubleshooting the Rules for the Interface Manager Distributor

The **show imds rules** command is used to display information for the IMD rules, which is a summary of basic information cached in the IMD. The **show** commands for the IM are used to display detailed data information on the rules.





## CHAPTER 4

# Troubleshooting Router Switch Fabric

This chapter describes techniques that you can use to troubleshoot router switch fabric. It includes the following sections:

- [CRS-1 Switch Fabric Overview, page 4-111](#)
- [Understanding the Flags Field, page 4-113](#)
- [Using the Online Diagnostics Tools, page 4-115](#)
- [Verifying and Troubleshooting the Fabric Plane State, page 4-115](#)
- [Verifying and Troubleshooting Up Fabric Planes, page 4-121](#)
- [Troubleshooting Down Fabric Planes, page 4-126](#)
- [Guidelines for Maintenance of Fabric Links, page 4-130](#)

## CRS-1 Switch Fabric Overview

The switch fabric is constructed of eight independent fabric planes. Each plane consists of at least three stage switch element ASICs (SEAs), which are collectively known as *switch fabric elements* (SFE). While all SEAs are identical, the operational behavior of a SEA is defined according to the position of the SEA in the fabric. A SEA set for Stage 1 operation is known as an S1 ASIC. A SEA set for Stage 2 operation is known as an S2 ASIC. A SEA set for Stage 3 operation is known as an S3 ASIC. The complete fabric is composed of the ingress queue ASIC on the ingress modular services card (MSC) or designated route processor (D)RP, the S1, S2, and S3 ASICs on the switch fabric planes and one of more fabric queue ASICs on the destination MSC or (D)RP.



### Note

For Cisco CRS-1 Multishelf Systems, connect to the route processor (RP) for the owner logical router (LR) when troubleshooting the fabric plane. When you are connected to the owner LR, you have control over the entire system and all cards assigned to the owner LR.

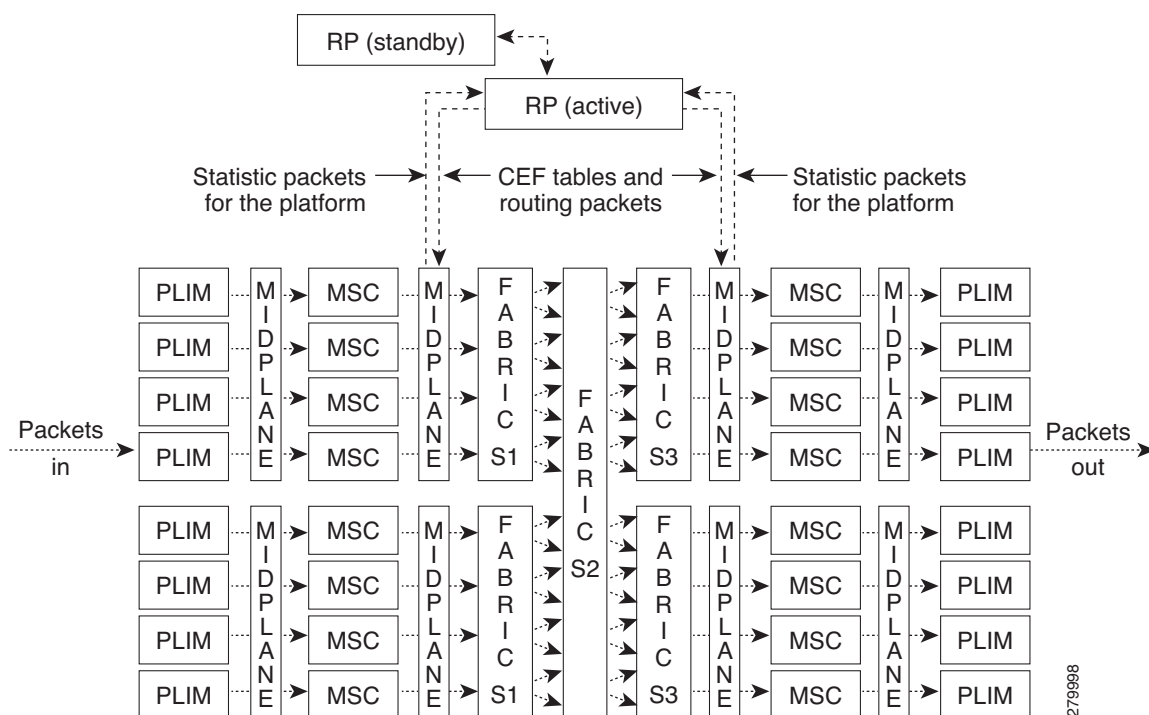


### Note

Line cards in Cisco CRS-1s are called modular services cards (MSCs).

[Figure 4-1](#) shows the architecture of the Cisco CRS-1. It illustrates the flow of packets from the ingress PLIM, through the fabric (Stages 1, 2, and 3) and out the egress PLIM.

Figure 4-1 Cisco CRS-1 Architecture



Between ingress queue, S1, S2, S3, and fabric queue ASICs are communication paths known as *fabric links*. An individual fabric link has a bandwidth capacity of 2.5 Gbps. A link is unidirectional.

An ingress MSC has multiple 2.5 Gbps links from the ingress queue ASIC on the MSC to an individual fabric plane and to an individual S1 ASIC on that fabric plane. Connections are provided in the same fashion to each of the eight fabric planes that are present in a system in normal operation.

All S1 ASICs have at least one 2.5 Gbps link to every S2 ASIC present on that fabric plane. An S1-to-S2 link does not and cannot span across fabric planes.

All S2 ASICs have at least one 2.5 Gbps link to every S3 ASIC present on that fabric plane. An S2-to-S3 link does not and cannot span across fabric planes.

An egress MSC has multiple 2.5 Gbps links from the S3 ASIC to one or two fabric queue ASICs present on that MSC.

In a 16-slot single-chassis system, the eight switch fabric planes are contained on eight individual fabric cards known as FC/S or S123. Each of the fabric cards contains two S1, two S2, and four S3 ASICs.

In an 8-slot system, the eight switch fabric planes are contained on four individual fabric cards known as FC/S or HS123. Each fabric card contains two planes. Each plane is composed of one S1, one S2, and one S3 ASIC.

In a 4-slot system, there are four fabric planes. The fabric planes are contained on four individual fabric cards known as FC/s or QS123. Each plane is composed of one S1, one S2, and one S3 ASIC.

Each S13 card contains two S1 and four S3 ASICs and corresponds to one switch fabric plane. Each fabric card chassis has one or more switch fabric planes and each fabric plane consists of one or more S2 fabric cards. Each S2 fabric card contains six S2 ASICs.

Data is not transmitted across the fabric in packets. Packets are segmented into cells by the ingress queue ASIC and reassembled back to packets by the fabric queue ASIC. After a cell is placed on a plane, it will remain on that plane until it reaches the egress MSC. While the system is operational, cells are always being sent and received even though they may not contain any data. Cells with no data are termed as *idle cells*.

Cell forwarding on the S3 ASIC on both the Cisco CRS-1 4-slot, 8-slot, and 16-slot systems involves examination of the cell header. Using the cell header, the S3 is able to determine which fabric queue ASIC the cells should be sent to. The S3 ASIC injects cells in a round-robin fashion across the multiple links connecting the S3 ASIC to the fabric queue ASIC.

As the cell is created, forward error correction (FEC) data is calculated and appended. The FEC code is used on the transmission links to ensure the integrity of the data as it is transmitted and received. The FEC data is removed, checked, recalculated, and reapplied by each SFE in the path until it is delivered to the egress MSC. The FEC value is checked before a cell is processed by the SFE. If an error is detected, the FEC code can be used to recover the cell. Depending on the nature of the error, recovery may not be possible, in which case, the cell is discarded.

A system can operate with seven planes with out performance degradation. As further planes are removed, overall capacity degrades but the system remains operational. A system requires a minimum of two planes to maintain service. One plane must be odd numbered and the other plane must be even numbered.

**Note**

A 4-slot system operates on four planes. The loss of a plane reduces the usable forwarding capacity by approximately 6Gbps.

A plane marked MCAST_DOWN means that as far as the ingress line cards are concerned, not all destination FabricQ ASICs can be reached via the plane. Since the destinations of multicast traffic cannot be predetermined due to their dynamic nature, the plane is termed as MCAST_DOWN. Multicast traffic will use the other fabric planes to deliver data to the appropriate egress line cards. Unicast traffic will continue to use the plane if traffic is destined to a FabricQ ASIC that is still reachable. If traffic is destined for a FabricQ ASIC that is not unreachable, the other available plane is used instead of the plane in the MCAST_DOWN state.

## Understanding the Flags Field

A Flags field is provided in some fabric-related **show** commands. The flag is an abbreviated reason for a link being placed in the down state. [Table 4-1](#) provides descriptions of the flags.

**Table 4-1 Fabric show Command Flags**

<i>P - plane admin down</i> —plane has been taken out of service by administrative action	<i>p - plane oper down</i> —plane is not able to operate or has been taken out of service by administrative action
<i>C - card admin down</i> —card has been taken out of service by administrative action (reserved for future use)	<i>c - card oper down</i> —flag is set in response to system notification that a card will soon be powered off. This is a transient condition lasting a few seconds before power is withdrawn. Unusual to see this flag in practice

**Table 4-1 Fabric show Command Flags (continued)**

<i>L - link port admin down</i> —link has been taken out of service by administrative action	<i>l - linkport oper down</i> —flag is set or cleared in response to messages from the ASIC driver code. Flags are enabled when the ASIC is first discovered. Flag is set if the ASIC driver has signaled that an individual link port is not usable for fabric traffic, or that the driver has confirmed that the link is ready. Two common reasons for seeing an l flag that are initialization is not completed yet or that sufficient errors have been seen at a receive port so that the ASIC driver shuts the port down
<i>A - ASIC admin down</i> —ASIC has been taken out of service by administrative action. If the A flag is set, it is seen on all links connected to that ASIC	<i>a - ASIC oper down</i> —flag is set or cleared in response to messages from the ASIC driver code. Flags are enabled when the ASIC is first discovered. Flag is set if the ASIC driver has signaled that an SEA ASIC is not usable for fabric traffic. If the a flag is set, it is seen on all links connected to that ASIC
<i>B - bundle port admin down</i> —bundle port has been taken out of service by administrative action. If the B flag is set, it is seen on all links associated to that bundle	<i>b - bundle port oper down</i> —bundle port cannot operate or has been taken out of service by administrative action. If the b flag is set, it is seen on all links associated to that bundle
<i>I - bundle admin down</i> —bundle has been taken out of service by administrative action	<i>i - bundle oper down</i> —bundle cannot operate or has been taken out of service by administrative action
<i>N - node admin down</i> —Node has been taken out of service by administrative action	<i>n - node down</i> —node cannot operate or has been taken out of service by administrative action
<i>o - other end of link down</i> —only applies to TX links. Indicates that receiver link port is down	<i>d - data down</i> —idle cells are being transmitted or received, but link is not able to transport data cells. Flag is set in response to a message from the ASIC driver. Link ports go into this state during link startup. It is typically transient and should only persist if something goes wrong with the startup process
<i>f - failed component downstream</i> —flag is set by fabric status database if it decides to stop using a link because of some downstream failure	
<i>m - plane multicast down</i> —not all fabric queue ASICs present on the plane are reachable; multicast forwarding is not operational on this plane. Unicast traffic is unaffected.	

## Using the Online Diagnostics Tools

The online diagnostics tools can alert you to potential problems in the optical connections between the S2 fabric cards in the Fabric Card Chassis and the S1/S3 fabric cards in the Line Card Chassis. For information on using the online diagnostics, see the [“Using Diagnostic Commands” section on page 7-166](#).

## Verifying and Troubleshooting the Fabric Plane State

To verify and troubleshoot the fabric plane state, perform the following procedure.

**Note**

All fabric troubleshooting should be performed in administration executive (admin EXEC) mode while you are logged into the default logical router. This allows you to view system-wide parameters.

### SUMMARY STEPS

1. **admin**
2. **show controllers fabric plane all detail**
3. **show controllers fabric plane all statistics**
4. **show controllers fabric plane *plane_id* statistics detail**
5. **show controllers fabric link health**
6. **show controllers fabric sfe s1 all | include UP.*DOWN**  
**show controllers fabric sfe s2 all | include UP.*DOWN**  
**show controllers fabric sfe s3 all | include UP.*DOWN**  
**show controllers fabric sfe ingress all | include UP.*DOWN** (for MSC fabric interface ASICs)  
**show controllers fabric sfe fabricq all | include UP.*DOWN** (for MSC fabric interface ASICs)
7. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>admin</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin	Enters administration executive (admin EXEC) mode.
Step 2	<b>show controllers fabric plane all detail</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all	<p>Displays system fabric plane information from all fabric planes.</p> <p>Check the number of planes in operation. If there are any planes that are down (either the Admin State or Oper State is DOWN), proceed to the <a href="#">“Troubleshooting Down Fabric Planes”</a> section on page 4-126.</p> <p>A system can operate with seven planes without performance degradation. As further planes are removed, overall capacity degrades, but the system remains operational. A system must have a minimum of two operational planes to maintain service. One must be an odd-numbered plane, the other must be an even-numbered plane.</p>
Step 3	<b>show controllers fabric plane all statistics</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all statistics	<p>Displays controller fabric statistics for all planes.</p> <p>Errors are indicated in the last three columns in the output:</p> <ul style="list-style-type: none"> <li>• CE=correctable errors</li> <li>• UCE=uncorrectable errors</li> <li>• PE=parity errors (in an ASIC memory)</li> </ul> <p>The Out Cells value should be equal or greater than the In Cells value because of multicast copies created in the fabric. If the Out Cells value is equal to or greater than the In Cells value, proceed to <a href="#">Step 4</a>.</p> <p>If the values in the Out Cells column are much less than the values in the In Cells column, cells are being dropped. If the Out Cells value is less than the In Cells value, proceed to the <a href="#">“Verifying and Troubleshooting Up Fabric Planes”</a> section on page 4-121.</p> <p>If the CE, UCE or PE values are incrementing, contact Cisco Technical Support. For contact information for Cisco Technical Support, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.</p>

	Command or Action	Purpose
Step 4	<b>show controllers fabric plane <i>plane_id</i> statistics detail</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric plane 0 statistics detail	<p>Displays controller fabric statistics for a specific plane.</p> <p>The output is displayed for unicast and multicast. This allows you to determine if a unicast problem was hidden by multicast cells. If the Total received unicast data cells value is much lower than the Total transmitted unicast data cells, there are transmission problems. If there is a large drop in the number of Total transmitted unicast data cells compared to the Total received unicast data cells, proceed to the <a href="#">“Verifying and Troubleshooting Up Fabric Planes”</a> section on page 4-121.</p> <p>The Total unicast lost cells and Total multicast lost cells displays the number of cells that were received by an ASIC but could not be handled. Dropped multicast cells can be due to congestion in the fabric. Dropped unicast cells indicate a more serious problem. If dropped unicast cells are detected, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.</p>
Step 5	<b>show controllers fabric link health</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric link health	<p>Displays the number of links that are operationally up in each plane.</p> <p>When the minimum required threshold is exceeded for the LCC the system changes the status of the plane to MCAST_DOWN.</p> <p>If connectivity is lost to all LCC the system changes the status of the plane to DOWN.</p>

	Command or Action	Purpose
Step 6	<pre>show controllers fabric sfe s1 all   include UP.*DOWN show controllers fabric sfe s2 all   include UP.*DOWN show controllers fabric sfe s3 all   include UP.*DOWN show controllers fabric sfe ingress all   include UP.*DOWN show controllers fabric sfe fabricq all   include UP.*DOWN</pre> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe s1 all   include UP.*DOWN RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe s2 all   include UP.*DOWN RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe s3 all   include UP.*DOWN RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe ingress all   include UP.*DOWN RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe fabricq all   include UP.*DOWN</pre>	<p>Use these commands on the ASICs in the affected planes to help isolate the problem to a specific board.</p> <p>Once you have isolated the problem to a specific board, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.</p>
Step 7	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The following example shows how to confirm the number of fabric planes.

Verify the number of fabric planes in operation:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail
```

```
Flags: P - plane admin down,      p - plane oper down
      C - card admin down,        c - card oper down
      L - link port admin down,    l - linkport oper down
      A - asic admin down,         a - asic oper down
      B - bundle port admin Down,  b - bundle port oper down
      I - bundle admin down,       i - bundle oper down
      N - node admin down,         n - node down
      o - other end of link down   d - data down
      f - failed component downstream
      m - plane multicast down
```

Plane Id	Admin State	Oper State	Down Flags	Total Bundles	Down Bundles
0	UP	UP		9	3
1	UP	UP		9	3
2	UP	UP		9	3
3	UP	UP		9	3
4	UP	UP		9	3
5	UP	UP		9	3
6	UP	UP		9	3
7	UP	UP		9	3

The output above shows all eight fabric planes in the up state.



The following output shows Fabric Plane 1 marked as MCAST_DOWN:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail
```

```
Flags: P - plane admin down,      p - plane oper down
       C - card admin down,       c - card oper down
       L - link port admin down,   l - linkport oper down
       A - asic admin down,       a - asic oper down
       B - bundle port admin Down, b - bundle port oper down
       I - bundle admin down,     i - bundle oper down
       N - node admin down,       n - node down
       o - other end of link down d - data down
       f - failed component downstream
       m - plane multicast down
```

Plane Id	Admin State	Oper State	Down Flags	Total Bundles	Down Bundles
0	UP	UP		0	0
1	UP	MCAST_DOWN	m	0	0
3	UP	UP		0	0
4	UP	UP		0	0
5	UP	UP		0	0
6	UP	UP		0	0
7	UP	UP		0	0

The fabric plane is operational, but not all destination fabric queue application-specific integrated circuit (ASICs) can be reached through this fabric plane, as far as the ingress modular services cards (MSCs) are concerned. Because destinations of multicast traffic cannot be predetermined due to their dynamic nature, the fabric plane is shown as MCAST_DOWN. Multicast traffic uses the other fabric planes to deliver data to the appropriate egress MSCs. Unicast traffic continues to use the fabric plane if traffic is destined to a fabric queue ASIC that is still reachable. If traffic is destined for a fabric queue ASIC that is deemed unreachable, the other available fabric plane is used instead of the plane in the MCAST_DOWN state. To change the operational state of a fabric plane from MCAST_DOWN to up, contact Cisco Technical Support. See the [“Obtaining Documentation and Submitting a Service Request” section on page viii](#) in the [Preface](#).

The following output shows Fabric Plane 0 as administratively shut down:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail
```

```
Flags: P - plane admin down,      p - plane oper down
       C - card admin down,       c - card oper down
       L - link port admin down,   l - linkport oper down
       A - asic admin down,       a - asic oper down
       B - bundle port admin Down, b - bundle port oper down
       I - bundle admin down,     i - bundle oper down
       N - node admin down,       n - node down
       o - other end of link down d - data down
       f - failed component downstream
       m - plane multicast down
```

Plane Id	Admin State	Oper State	Down Flags	Total Bundles	Down Bundles
0	DOWN	DOWN	P	0	0
1	UP	UP		0	0
2	UP	UP		0	0
3	UP	UP		0	0
4	UP	UP		0	0
5	UP	UP		0	0
6	UP	UP		0	0
7	UP	UP		0	0

The following output shows Fabric Plane 4 as administratively shutdown and powered off:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail
```

```
Flags: P - plane admin down,      p - plane oper down
       C - card admin down,       c - card oper down
       L - link port admin down,  l - linkport oper down
       A - asic admin down,       a - asic oper down
       B - bundle port admin Down, b - bundle port oper down
       I - bundle admin down,     i - bundle oper down
       N - node admin down,       n - node down
       o - other end of link down d - data down
       f - failed component downstream
       m - plane multicast down
```

Plane Id	Admin State	Oper State	Down Flags	Total Bundles	Down Bundles
0	UP	UP		0	0
1	UP	UP		0	0
2	UP	UP		0	0
3	UP	UP		0	0
4	DOWN	DOWN	pPm	0	0
5	UP	UP		0	0
6	UP	UP		0	0
7	UP	UP		0	0

The following example shows how to display statistics for all fabric planes:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all statistics
```

Plane	In Cells	Out Cells	CE Cells	UCE Cells	PE Cells
0	736182590951	736183103374	0	0	0
1	736324880091	736325098791	0	0	0
2	736315213586	736315442992	0	0	0
3	736299535716	736299764252	0	0	0
4	736300399513	736300627048	0	0	0
5	736295116556	736295346246	0	0	0
6	736311177372	736311406904	0	0	0
7	736296917574	736297149734	0	0	0

CE = Errored cell with a correctable error detected using FEC code

UCE = Errored cell with an uncorrectable error detected using FEC code

PE = Parity error present within a cell when processed by SFE

The following example shows to display detailed fabric plane statistics for a specified fabric plane:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane 0 statistics detail
```

```
The fabric plane number is 0
Total number of providers for the statistics: 1
Total received data cells: 749351837428
Total received unicast data cells: 749351660427
Total received multicast data cells: 177001
Total transmitted data cells: 749352358340
Total transmitted unicast data cells: 749351671469
Total transmitted multicast data cells: 686871
Total received correctable errored cells: 0
```

```

Total received uncorrectable errored cells: 0
Total received parity error cells: 0
Total unicast lost cells: 0
Total multicast lost cells: 0
Last clearing of "show controller fabric plane" counters never

```

The following example shows how to display the health of the fabric links, including the number of links that are operationally up between the fabric stages.

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric link health
```

```

Wed May  2 16:29:49.102 EST EDT
Flags: P - plane admin down,      p - plane oper down
      C - card admin down,        c - card  oper down
      A - asic admin down,        a - asic  oper down
      L - link port admin down,    l - linkport oper down
      B - bundle port admin Down,  b - bundle port oper down
      I - bundle admin down,       i - bundle oper down
      N - node admin down,         n - node  down
      X - ctrl admin down,         x - ctrl  down
      o - other end of link down   d - data  down
      f - failed component downstream
      m - plane multicast down,    s - link port permanently shutdown
      t - no barrier input         O - Out-Of-Service oper down
      T - topology mismatch down  e - link port control only

```

Mismatched Plane details

```

-----
Plane Admin Oper up->dn      Down      Total      Down
Id      State State counter    Flags    Bundles    Bundles
-----

```

Link Usage Summary

```

-----
                                     # of OPER UP Links
Rack      Plane Group   Min      Max      Current
Num      stage-stage Num  Num  Required Available Available
-----
0         S1-S2       0    0    1         36         36
0         S2-S3       0    0    49        72         72
0         S1-S2       0    1    1         36         36

```

Mismatched Link detail

```

-----
                                     FSDB/ FSDB/
                                     Drvr/ Drvr/
Sfe Port                                     Bport
R/S/M/A/P                                     Num
                                     Avail Oper Down Plane Other Local/
                                     State State Flags Num  End  Remote
-----

```

## Verifying and Troubleshooting Up Fabric Planes

To verify and troubleshoot the up fabric planes, perform the following procedure.

For Cisco CRS-1 Multishelf Systems the S1 to S2 links and the S2 to S3 links are inter-chassis links.

For Cisco CRS-1 single-shelf systems the links are internal chassis links. The commands used are the same.






## SUMMARY STEPS

1. **admin**
2. **show controllers fabric link port s2rx all | include UP.*DOWN.*SM**  
**show controllers fabric link port s3rx all | include UP.*DOWN.*SM**  
**show controllers fabric link port s2tx all | include UP.*DOWN.*SM**  
**show controllers fabric link port s3tx all | include UP.*DOWN.*SM**
3. **show controllers fabric link port s2rx all statistics | exclude \ 0 +0 +0\$**  
**show controllers fabric link port s3rx all statistics | exclude \ 0 +0 +0\$**
4. **config**
5. **controllers fabric link port {s1rx | s1tx | s2rx | s2tx | s3rx | s3tx} location link-id shutdown**
6. Reload the fabric card.
7. Clean and reconnect the fiber-optic connectors (Cisco CRS-1 Multishelf Systems).

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>admin</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin	Enters administration executive (admin EXEC) mode.
Step 2	<b>show controllers fabric link port s2rx all   include UP.*DOWN.*SM</b> <b>show controllers fabric link port s3rx all   include UP.*DOWN.*SM</b> <b>show controllers fabric link port s2tx all   include UP.*DOWN.*SM</b> <b>show controllers fabric link port s3tx all   include UP.*DOWN.*SM</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s2rx all   include UP.*DOWN.*SM RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s3rx all   include UP.*DOWN.*SM RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s2tx all   include UP.*DOWN.*SM RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s3tx all   include UP.*DOWN.*SM	Displays the link state of a fiber port. The command output shows interchassis link ports that are operationally down, excluding those that are administratively down and those that are completely disconnected.  Repeat this command for each stage and direction and check if a fabric port link is down (ADMIN UP and OPER DOWN).  The following down flags may be displayed: <ul style="list-style-type: none"> <li>• l—Link is down</li> <li>• o—Downed link port due to the other end of the link being down</li> <li>• a—ASIC is down</li> <li>• A or L—ASIC (A) or link (L) is administratively shut down</li> </ul> If the link state of a fabric port is down, proceed to <a href="#">Step 5</a> to shut down the link.

	Command or Action	Purpose
Step 3	<pre>show controllers fabric link port s2rx all statistics   exclude \ 0 +0 +0\$ show controllers fabric link port s3rx all statistics   exclude \ 0 +0 +0\$</pre> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(admin)# show control fabric link port s2rx all statistics   exclude \ 0 +0 +0\$ RP/0/RP0/CPU0:router(admin)# show control fabric link port s3rx all statistics   exclude \ 0 +0 +0\$</pre>	<p>Displays any link that has an error count, indicating a problem link even if the link has not been shut down.</p> <p>If there are UCE errors on a link, shut down the link to avoid traffic loss. Proceed to <a href="#">Step 5</a> to shut down the link.</p>
Step 4	<pre>config</pre> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(admin)# config</pre>	Enters administration configuration mode.
Step 5	<pre>controllers fabric link port {s1rx   s1tx   s2rx   s2tx   s3rx   s3tx} location link-id shutdown</pre> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(admin-config)# controllers fabric link port s1rx 0/0/cpu0 1 shutdown</pre>	<p>Places a link out of service. This command closes both ends of the link.</p> <p>Note that performing such an operation should be conducted in conjunction with Cisco Technical Support engineers. For Cisco Technical Support contact information, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the Preface.</p> <p>If you have placed a link out of service on a Cisco CRS-1 single-shelf system, proceed to <a href="#">Step 6</a>.</p> <p>If you have placed a link out of service on a Cisco CRS-1 Multishelf System, proceed to <a href="#">Step 7</a>.</p>
Step 6	Reload the fabric card containing the out of service link.	<p>Perform this step if you have placed a fabric link out of service on a Cisco CRS-1 single-shelf system.</p> <p>See the “Managing the Router Hardware” chapter of <i>Cisco IOS XR System Management Configuration Guide</i> for information on reloading the fabric card.</p> <p>If the problem is not resolved, Cisco Technical Support. For Cisco Technical Support contact information, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the Preface.</p>

Command or Action	Purpose
<b>Step 7</b> Clean and reconnect the fiber-optic connectors for the out of service link.  Review these reminders and warnings before inspecting or handling your fiber-optic connection:	Perform this step if you have placed a link out of service on a Cisco CRS-1 Multishelf System.  You must clean the fabric fiber-optic connectors with the specially designed Cisco kit (CRS-FIBER-CLN-KIT=) and you must use the procedure in the <i>Cisco CRS-1 Carrier Routing System Fiber-Optic Cleaning Guide</i> . This document is available on the Maintain and Operate documentation site: <a href="http://www.cisco.com/en/US/products/ps5763/prod_maintenance_guides_list.html">http://www.cisco.com/en/US/products/ps5763/prod_maintenance_guides_list.html</a> .
 <b>Warning</b> Because invisible radiation may be emitted from the aperture of the port when no fiber cable is connected, avoid exposure to radiation and do not stare into open apertures. Statement 125	 <b>Caution</b> If you do not use the specified kit and procedure, the optical connectors will not be cleaned properly and interruption of the signal will continue to happen.
 <b>Warning</b> Class 1M laser radiation when open. Do not view directly with optical instruments. Statement 281	If cleaning and reconnecting the optical fibers does not resolve the down link, you might need to replace the optical fibers. However, contact Cisco Technical Support first to discuss further troubleshooting options.
 <b>Warning</b> Laser radiation. Do not view directly with optical instruments. Class 1M laser product. Statement 283	For Cisco Technical Support contact information, see the “Obtaining Documentation and Submitting a Service Request” section on page viii in the Preface.
 <b>Warning</b> For diverging beams, viewing the laser output with certain optical instruments within a distance of 100 mm may pose an eye hazard. For collimated beams, viewing the laser output with certain optical instruments designed for use at a distance may pose an eye hazard. Statement 282	

## Examples

The following example shows how to display the link state of a fiber port. The output displays link ports that are operationally down, excluding those that are administratively down and those that are completely disconnected.

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s2rx all statistics
```

```
Wed May 2 16:29:49.102 EST EDT
```

```
Total racks: 1
```

```
Rack 0:
```

UCE	SFE Port PE R/S/M/A/P Cells	In Data Cells	In Idle Cells	CE Cells
0/SM0/SP/0/0	9149278	5437189623455	0	

```

0      0
0      0/SM0/SP/0/1      9149418      5438017457030      0
0      0
0      0/SM0/SP/0/2      9149418      5438017379235      0
0      0
0      0/SM0/SP/0/3      9149418      5438017274614      0
0      0
0      0/SM0/SP/0/4      9149418      5438017189053      0
0      0
0      0/SM0/SP/0/5      9149418      5438017086116      0
0      0
0      0/SM0/SP/0/6      9149418      5438016991625      0
0      0
0      0/SM0/SP/0/7      9149418      5438016903486      0
0      0
0      0/SM0/SP/0/8      9149418      5438016817714      0
0      0
.
.
.

```

The following example shows how to display a link with an error count, indicating a problem link even if the link has not been shut down.

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric link port s2rx all statistics |
exclude \ 0 +0 +0$
```

Total racks: 1

Rack 0:

SFE Port R/S/M/A/P	In Data Cells	In Idle Cells	CE Cells	UCE Cells	PE Cells
-----					

The following example shows how to place a link out of service.



#### Note

Performing this operation should be conducted in conjunction with Cisco Technical Support engineers.

```
RP/0/RP0/CPU0:router(admin-config)# controllers fabric link port slrx all brief
```

Wed May 2 16:31:45.560 EST EDT

```

Flags: P - plane admin down,      p - plane oper down
       C - card admin down,       c - card oper down
       L - link port admin down,  l - linkport oper down
       A - asic admin down,       a - asic oper down
       B - bundle port admin Down, b - bundle port oper down
       I - bundle admin down,     i - bundle oper down
       N - node admin down,       n - node down
       o - other end of link down d - data down
       f - failed component downstream
       m - plane multicast down

```

Sfe Port end R/S/M/A/P	Admin State	Oper State	Down Flags	Other End	Near-end Bport	Far- Bport
-----						
0/SM0/SP/0/0	UP	UP		0/SM0/SP/0/0		
0/SM0/SP/0/1	UP	UP		0/SM0/SP/0/2		
0/SM0/SP/0/2	UP	UP		0/SM0/SP/0/4		
0/SM0/SP/0/3	UP	UP		0/SM0/SP/0/6		
0/SM0/SP/0/4	UP	UP		0/SM0/SP/0/8		

0/SM0/SP/0/5	UP	UP	0/SM0/SP/0/10
0/SM0/SP/0/6	UP	UP	0/SM0/SP/0/12
0/SM0/SP/0/7	UP	UP	0/SM0/SP/0/14
0/SM0/SP/0/8	UP	UP	0/SM0/SP/0/16
0/SM0/SP/0/9	UP	UP	0/SM0/SP/0/18
0/SM0/SP/0/10	UP	UP	0/SM0/SP/0/20
0/SM0/SP/0/11	UP	UP	0/SM0/SP/0/22
0/SM0/SP/0/12	UP	UP	0/SM0/SP/0/24
0/SM0/SP/0/13	UP	UP	0/SM0/SP/0/26
.			
.			
.			

## Troubleshooting Down Fabric Planes

An individual link placed in a down state can be sustained without impact to the overall operation or forwarding capacity of the Cisco CRS-1 Multishelf System. At each stage of the fabric, there is a greater bandwidth capacity than can actually be generated by the modular services cards (MSCs). The egress side of the switch fabric has at least double the capacity of the ingress.

The loss of a single link reduces the switch fabric capacity on a single plane by 2.5 Gbps or approximately 1.5625 Gbps when considering 8b/10b encoding and cell tax overheads. The actual effective loss of bandwidth varies depending on where the link loss takes place. If a link were to be lost between the ingress queue and S1 stage or S1 and S2 stage, the reduction is against a capacity of 80 Gbps or 50 Gbps effective bandwidth. A link loss between S2 and S3 stages or S3 and fabric queue stages means that the reduction is against a capacity of 160 Gbps or 100 Gbps effective bandwidth.

Consider a situation in which a set of S1 links was lost on a single plane. Six or seven links would need to be taken out of service in order to reduce the effective bandwidth capacity by approximately 10 Gbps. It is, therefore, possible to contend that because the MSC is capable of transmitting at 40 Gbps, there is still sufficient capacity to carry the traffic without loss.

The system is designed to operate with seven out of eight fabric planes in operation and provide full line-rate forwarding capability, because it provides approximately 42 Gbps to each MSC. If another fabric plane is removed, forwarding capacity would be reduced to approximately 34 Gbps.

### SUMMARY STEPS

1. **admin**
2. **show controllers fabric plane all detail**
3. Bring up the plane:
  - a. **configure**
  - b. **no controllers fabric plane *plane-id* shutdown**
  - c. **end**
4. **show running-config**
5. **show controllers fabric connectivity all**
6. **show controllers fabric sfe {s1 | s2 | s3 | ingressq | fabricq} all**
7. **show controllers fabric bundle port all**



## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>admin</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin	Enters administration executive (admin EXEC) mode.
Step 2	<b>show controllers fabric plane all detail</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail	Displays information about the fabric plane.  Verify that the down plane is not administratively down. If the Admin State is down, the operational state should also be down. <ul style="list-style-type: none"> <li>If the plane is administratively down and the only flag is P, bring the plane up. Proceed to <a href="#">Step 3</a> to bring up the plane. See the “<a href="#">Understanding the Flags Field</a>” section on page 4-113 for detailed information on flags.</li> <li>If the plane is administratively up but the flag is m, the plane is multicast down and some destinations (for example, MSCs and route processors) are having connectivity issues to the fabric on that plane. Proceed to <a href="#">Step 5</a>. See the “<a href="#">Understanding the Flags Field</a>” section on page 4-113 for detailed information on flags.</li> <li>If the plane is administratively down and the flag is p, the plane is operationally down. Proceed to <a href="#">Step 6</a>. See the “<a href="#">Understanding the Flags Field</a>” section on page 4-113 for detailed information on flags.</li> </ul>
Step 3	<b>configure</b> <b>no controllers fabric plane</b> <i>plane-id</i> <b>shutdown</b> <b>end</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)#configure RP/0/RP0/CPU0:router(admin_config)# no controllers fabric plane 3 shutdown RP/0/RP0/CPU0:router(admin_config)# end	The following tasks must be completed to bring up the plane: <ul style="list-style-type: none"> <li><b>configure</b>—Enters administration configuration mode</li> <li><b>no controllers fabric plane</b> <i>plane-id</i> <b>shutdown</b>—Performs a graceful shutdown of the fabric plane to ensure that data is no longer flowing through the plane before a fabric reconfiguration or fabric plane migration</li> <li><b>end</b>—Saves the configuration change</li> </ul>
Step 4	<b>show running-config</b>  <b>Example:</b> RP/0/RP0/CPU0:router(admin)# show running-config	For Cisco CRS-1 Multishelf Systems only.  Displays the contents of the administrative running configuration.  Verify that the plane is set. In the Cisco CRS-1 Multishelf System, the command output indicates which fiber module in the fabric card chassis (FCC) is connected from the plane.  The following is an example of the output:  <pre>controllers fabric plane 0 topology single-module location F0/SM4/FM</pre>

Command or Action	Purpose
<p><b>Step 5</b>    <code>show controllers fabric connectivity all</code></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router(admin)# show controllers fabric connectivity all</p>	<p>Displays controller fabric connectivity information for all fabric ports.</p> <p>For the down plane, check if the MSCs and route processors have receive and transmit connectivity to the fabric.</p> <ul style="list-style-type: none"> <li>• If there is no transmit and receive connectivity to the fabric, contact Cisco Technical Support. For Cisco Technical Support contact information, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the Preface.</li> <li>• If there is no transmit and receive connectivity to a MSC that is not required, remove it from the system or power it down. This will bring the plane back into the up state. See <i>Installing the Cisco CRS-1 Carrier Routing System 16-Slot Line Card Chassis</i> at the following URL for information on removing MSCs:  <a href="http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html">http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html</a></li> </ul>
<p><b>Step 6</b>    <code>show controllers fabric sfe {s1   s2   s3   ingressq   fabricq} all</code></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router(admin)# show controllers fabric sfe s1 all</p>	<p>Displays ASIC information.</p> <p>Verify that all the ASICs are in the administrative and operational up state.</p> <ul style="list-style-type: none"> <li>• If all ASICs are in the up state, proceed to <a href="#">Step 7</a>.</li> <li>• If all the ASICs are not in the up state, contact Cisco Technical Support. For Cisco Technical Support contact information, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the Preface.</li> </ul>
<p><b>Step 7</b>    <code>show controllers fabric bundle port all</code></p> <p><b>Example:</b>  RP/0/RP0/CPU0:router(admin)# show controllers fabric bundle port all</p>	<p>For Cisco CRS-1 Multishelf Systems only.</p> <p>Displays bundle port information.</p> <p>Verify that all bundle ports are in the operational up state.</p> <ul style="list-style-type: none"> <li>• If a bundle port is in the operational down state, contact Cisco Technical Support, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the Preface.</li> </ul> <p><b>Note</b>    For this release, three of the nine ports will be down because there is no third rack connected to the fabric card chassis. The down ports are 6, 7, and 8.</p> <ul style="list-style-type: none"> <li>• If a bundle port is in the administrative down state, proceed to <a href="#">Step 3</a> to bring up the plane.</li> </ul>

## Examples

The output from the `show controllers fabric plane all detail` command displays information on the fabric plane:

```
RP/0/RP0/CPU0:router(admin)# show controllers fabric plane all detail
```

```
Flags: P - plane admin down,          p - plane oper down
```

C - card admin down,                    c - card oper down  
 L - link port admin down,            l - linkport oper down  
 A - asic admin down,                a - asic oper down  
 B - bundle port admin Down,        b - bundle port oper down  
 I - bundle admin down,              i - bundle oper down  
 N - node admin down,                n - node down  
 o - other end of link down        d - data down  
 f - failed component downstream  
 m - plane multicast down

Plane Id	Admin State	Oper State	Down Flags	Total Bundles	Down Bundles
0	UP	UP		9	3
1	UP	UP		9	3
2	UP	DOWN	p	0	0
3	UP	DOWN	p	0	0
4	UP	MCAST_DOWN	m	9	3
5	UP	UP		9	3
6	UP	DOWN	p	0	0
7	UP	DOWN	p	0	0

The output from the **show controllers fabric bundle port all** command displays the status of the bundle ports:

RP/0/RP0/CPU0:router(admin)# **show controllers fabric bundle port all**

Flags: P - plane admin down,            p - plane oper down  
       C - card admin down,              c - card oper down  
       L - link port admin down,        l - linkport oper down  
       A - asic admin down,              a - asic oper down  
       B - bundle port admin Down,      b - bundle port oper down  
       I - bundle admin down,            i - bundle oper down  
       N - node admin down,              n - node down  
       o - other end of link down        d - data down  
       f - failed component downstream  
       m - plane multicast down

Bundle Port R/S/M/P	Admin State	Oper State
0/SM0/SP/0	UP	UP
0/SM0/SP/1	UP	UP
0/SM0/SP/2	UP	UP
0/SM1/SP/0	UP	UP
0/SM1/SP/1	UP	UP
0/SM1/SP/2	UP	UP
0/SM4/SP/0	UP	UP
0/SM4/SP/1	UP	UP
0/SM4/SP/2	UP	UP
0/SM5/SP/0	UP	UP
0/SM5/SP/1	UP	UP
0/SM5/SP/2	UP	UP
1/SM0/SP/0	UP	UP
1/SM0/SP/1	UP	UP
1/SM0/SP/2	UP	UP
1/SM1/SP/0	UP	UP
1/SM1/SP/1	UP	UP
1/SM1/SP/2	UP	UP
1/SM4/SP/0	UP	UP
1/SM4/SP/1	UP	UP
1/SM4/SP/2	UP	UP
1/SM5/SP/0	UP	UP

```

1 / SM5 / SP / 1      UP      UP
1 / SM5 / SP / 2      UP      UP
F0 / SM0 / FM / 0     UP      UP
.
.
.

```

## Guidelines for Maintenance of Fabric Links

You need to maintain the fabric links in your router to avoid the loss of fabric redundancy.

The failure of a single fabric link does not necessarily have a major impact on the ability of the system to process traffic. Even if one of the eight fabric planes goes down, the system can process traffic without degradation. You should take into account your overall bandwidth requirement through the fabric before you decide whether to perform maintenance. If your system is currently stable and processing traffic, we *do not* recommend that you risk destabilizing the system with unnecessary maintenance procedures. For example, in multichassis systems, it is not necessary to clean the intershelf fibers if a single fabric link goes down.



### Caution

The maintenance guidelines in this section are applicable to large installations in which the loss of a single fabric plane does not result in degradation of throughput. If you have a smaller system, bandwidth limitations in your system, or the failures occur on a specific SDR, you might need to take immediate action if a smaller number of links fail. Otherwise, your system could experience a serious degradation of throughput or a loss of traffic.

You should consider performing fabric link maintenance in any of the following scenarios:

- In a single chassis or multichassis system, if four or more links to or from a linecard are in a failure state.



### Note

For RPs and DRPs, maintenance or replacement might be required if one or two links fail, depending on whether these devices are installed in a CRS-1 4-slot, 8-slot, or 16-slot system.

- In a single chassis or multichassis system, if any plane has four or more links down across all its linecards.
- In a multichassis system, if any plane has four or more links down to any individual linecard chassis.

See the other sections in this chapter for additional information on CLI commands and fiberoptic cleaning procedures. If you are unsure whether you need to perform maintenance procedures on your system, contact Cisco Technical Support first to discuss your options. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on [page viii](#) in the [Preface](#).



## CHAPTER 5

# Troubleshooting Interfaces

---

This chapter describes techniques that you can use to troubleshoot interfaces. It includes the following sections:

- [Verifying and Troubleshooting Configured Interfaces, page 5-131](#)
- [Verifying and Troubleshooting Pluggable Optical Interfaces, page 5-139](#)
- [Troubleshooting Common Issues with Bundle Interfaces, page 5-144](#)

## Verifying and Troubleshooting Configured Interfaces

To troubleshoot the configured interfaces on Cisco IOS XR software, perform the following procedure.

### SUMMARY STEPS

1. **show interfaces** *type instance*
2. **show controllers** *interface-type interface-instance stats*
3. **show netio idb** *interface-type interface-instance*
4. **show controllers plim asic statistics interface** *type instance*  
or  
**show hw-module subslot** *address counters mac index*  
or  
**show hw-module subslot counters framer**
5. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show interfaces</b> <i>type instance</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces gigabitEthernet 0/0/0/0	Displays statistics for all interfaces configured on the specified node. Check for interface errors.
Step 2	<b>show controllers</b> <i>interface-type</i> <i>interface-instance</i> <b>stats</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers gigabitEthernet 0/0/0/0 stats	Displays interface controller status and configuration statistics for the specified node. Check for input drops.  If input drops are found, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.
Step 3	<b>show netio idb</b> <i>interface-type</i> <i>interface-instance</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show netio idb gigabitEthernet 0/0/0/0	Displays network input and output information for a specified node.  Check the software counters for the interface. Under Chains and Protocol chains in the output check if any drops occurred at a particular point in the Encap or Decap of the packet. The drops are displayed in the column on the right, showing drop packets and bytes for each step in the chain.
Step 4	<b>show controllers plim asic statistics</b> <i>interface</i> <i>type instance</i> or <b>show hw-module subslot</b> <i>address</i> <b>counters</b> <i>mac</i> <i>index</i> or <b>show hw-module subslot counters</b> <i>framer</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers plim asic statistics interface tenGigE 1/3/0/0 or RP/0/RP0/CPU0:router# show hw-module subslot 0/0/0 counters mac 0 or RP/0/RP0/CPU0:router# show hw-module subslot counters framer	<ul style="list-style-type: none"> <li>Use the <b>show controllers plim</b> command for fixed PLIMs. Check for interface drops on fixed physical layer interface modules (PLIMs).</li> <li>Use the <b>show hw-module subslot</b> <i>address</i> <b>counters</b> <i>mac</i> command and show hw-module subslot counters framer command for shared port adapters (SPAs). Check for interface drops on SPAs.</li> </ul> Check the application-specific integrated circuit (ASIC) counters for drop counters or error counters incrementing for the interface.
Step 5	Contact Cisco Technical Support.	For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

The following example shows POS 0/0/1/0 with no input drop counters.

```
RP/0/RP0/CPU0:router# show interfaces pos 0/0/1/0
```

```
POS0/0/1/0 is up, line protocol is up
Hardware is Packet over SONET/SDH
Internet address is 172.18.140.1/24
MTU 4474 bytes, BW 155520 Kbit
reliability 255/255, txload 1/255, rxload 1/255
```

```

Encapsulation HDLC, crc 32, controller loopback not set, keepalive set (10 sec)
Last clearing of "show interface" counters never
5 minute input rate 3000 bits/sec, 0 packets/sec
5 minute output rate 3000 bits/sec, 0 packets/sec
  199794 packets input, 222359750 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
      0 runts, 0 giants, 0 throttles, 0 parity
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  89911 packets output, 213413210 bytes, 0 total output drops
    0 output errors, 0 underruns, 0 applique, 0 resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions

```

RP/0/RP0/CPU0:router# **show controllers pos 0/0/1/0 framer statistics**

```

POS Driver Internal Cooked Stats Values for port 0
=====
Rx Statistics                                Tx Statistics
-----
Total Bytes:      0                        Total Bytes:      0
Good Bytes:       222379100                Good Bytes:       213436076
Good Packets:     199818                    Good Packets:     89927
Aborts:           0                        Aborts:           0
FCS Errors:       0                        Min-len errors:   0
Runts:            0                        Max-len errors:   0
FIFO Overflows:   0                        FIFO Underruns:   0
Giants:           0
Drops:            0

```

RP/0/RP0/CPU0:router# **show netio idb pos 0/0/1/0**

```

POS0/0/1/0 (handle: 0x010800a0, nodeid:0x1) netio idb:
-----
name:                POS0_0_1_0
interface handle:     0x010800a0
interface global index: 2
physical media type:  14
dchain ptr:           <0x482dd660>
echain ptr:           <0x48247d58>
fchain ptr:           <0x482dd774>
driver cookie:        <0x4824cd68>
driver func:          <0x4824cd54>
number of subinterfaces: 0
subblock array size:  0
DSNCF:                0x00000000
interface stats info:
  IN  unknown proto pkts: 0
  IN  unknown proto bytes: 0
  IN  multicast pkts: 0
  OUT multicast pkts: 0
  IN  broadcast pkts: 0
  OUT broadcast pkts: 0
  IN  drop pkts: 0
  OUT drop pkts: 0
  IN  errors pkts: 0
  OUT errors pkts: 0

```

Chains

```

-----
Base decap chain:
  hdlc                <14>  <0xfd6a0a74, 0x00000000>  <      0,      0>

```

Protocol chains:

-----

```

<Protocol number> (name) Stats
Type Chain_node <caps num> <function, context> <drop pkts, drop bytes>
<9> (chdlc) Stats IN: 48466 pkts, 3559516 bytes; OUT: 41378 pkts, 910312 bytes
Encap:
  l2_adj_rewrite <86> <0xfceada88, 0x482c390c> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>
Decap:
  queue_fifo <56> <0xfcedea4c, 0x482ddc30> < 0, 0>
  chdlc <13> <0xfd6a252c, 0x00000000> < 0, 0>
Fixup:
  l2_adj_rewrite <86> <0xfcead45c, 0x00000000> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>
<10> (clns) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
Encap:
  clns <15> <0xfcfaa030, 0x00000000> < 0, 0>
  hdlc <14> <0xfd6a0678, 0x00000000> < 0, 0>
  l2_adj_rewrite <86> <0xfceada88, 0x48305a90> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>
Decap:
  queue_fifo <56> <0xfcedea4c, 0x482ddc30> < 0, 0>
  clns <15> <0xfcfa9508, 0x00000000> < 0, 0>
Fixup:
  l2_adj_rewrite <86> <0xfcead45c, 0x00000000> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>
<12> (ipv4) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
Encap:
  ipv4 <26> <0xfd0f341c, 0x482dd460> < 0, 0>
  hdlc <14> <0xfd6a0678, 0x00000000> < 0, 0>
  l2_adj_rewrite <86> <0xfceada88, 0x48349b54> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>
Decap:
  queue_fifo <56> <0xfcedea4c, 0x482ddc30> < 0, 0>
  ipv4 <26> <0xfd0f3474, 0x00000000> < 0, 0>
Fixup:
  l2_adj_rewrite <86> <0xfcead45c, 0x00000000> < 0, 0>
  queue_fifo <56> <0xfcedea68, 0x482ddc30> < 0, 0>
  txm_nopull <60> <0xfce8fa5c, 0x482ddda4> < 0, 0>

```

Protocol SAFI counts:

-----

Protocol	SAFI	Pkts In	Bytes In	Pkts Out	Bytes Out
ipv4	Unicast	0	0	0	0
ipv4	Multicast	0	0	0	0
ipv4	Broadcast	0	0	0	0
ipv6	Unicast	0	0	0	0
ipv6	Multicast	0	0	0	0

The following example shows counters implemented for Multiprotocol Label Switching (MPLS) packets. The following output under Protocol Chains in the **show netio idb** command shows the MPLS packets incrementing:

```

mpls <25> <0xfcc7b2b8, 0x00000000> < 152, 17328>

```



```
RP/0/RP0/CPU0:router# show netio idb gigabitEthernet 0/2/0/1
```

```
GigabitEthernet0/2/0/1 (handle: 0x01280040, nodeid:0x21) netio idb:
```

```
-----
name: GigabitEthernet0_2_0_1
interface handle: 0x01280040
interface global index: 3
physical media type: 30
dchain ptr: <0x482e0700>
echain ptr: <0x482e1024>
fchain ptr: <0x482e13ec>
driver cookie: <0x4829fc6c>
driver func: <0x4829f040>
number of subinterfaces: 4096
subblock array size: 7
DSNCF: 0x00000000
```

```
interface stats info:
  IN unknown proto pkts: 0
  IN unknown proto bytes: 0
  IN multicast pkts: 0
  OUT multicast pkts: 0
  IN broadcast pkts: 0
  OUT broadcast pkts: 0
  IN drop pkts: 0
  OUT drop pkts: 0
  IN errors pkts: 0
  OUT errors pkts: 0
```

```
Chains
```

```
-----
Base decap chain:
  ether <30> <0xfd018cd8, 0x482c736c> < 0, 0>
```

```
Protocol chains:
```

```
-----
<Protocol number> (name) Stats
Type Chain_node <caps num> <function, context> <drop pkts, drop bytes>
<7> (arp) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
  Encap:
    l2_adj_rewrite <86> <0xfcaa997c, 0x4831a33c> < 0, 0>
    pcn_output <54> <0xfd054bfc, 0x48319f04> < 0, 0>
    q_fq <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
    txm_nopull <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
  Decap:
    pcn_input <55> <0xfd054bfc, 0x4830ba8c> < 0, 0>
    q_fq_input <96> <0xfd05f330, 0x48312c7c> < 0, 0>
    arp <24> <0xfcbfc2cc, 0x00000000> < 0, 0>
  Fixup:
    l2_adj_rewrite <86> <0xfcaa945c, 0x00000000> < 0, 0>
    pcn_output <54> <0xfd054bfc, 0x48319f04> < 0, 0>
    q_fq <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
    txm_nopull <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
<10> (clns) Stats IN: 0 pkts, 0 bytes; OUT: 1861623 pkts, 2062483853 bytes
  Encap:
    clns <15> <0xfcb2c80, 0x00000000> < 0, 0>
    ether <30> <0xfd0189b4, 0x482c736c> < 0, 0>
    l2_adj_rewrite <86> <0xfcaa997c, 0x482d8660> < 0, 0>
    pcn_output <54> <0xfd054bfc, 0x48319f04> < 0, 0>
    q_fq <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
    txm_nopull <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
  Decap:
    pcn_input <55> <0xfd054bfc, 0x4830ba8c> < 0, 0>
    q_fq_input <96> <0xfd05f330, 0x48312c7c> < 0, 0>
```

```

      clns                                <15> <0xfcbe2444, 0x00000000> < 0, 0>
Fixup:
  l2_adj_rewrite                         <86> <0xfcaa945c, 0x00000000> < 0, 0>
  pcn_output                             <54> <0xfd054bfc, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
<12> (ipv4)   Stats IN: 0 pkts, 0 bytes; OUT: 759095 pkts, 57691220 bytes
Encap:
  ipv4                                   <26> <0xfcc03dfc, 0x482e0414> < 0, 0>
  ether                                   <30> <0xfd0189b4, 0x482c736c> < 0, 0>
  l2_adj_rewrite                         <86> <0xfcaa997c, 0x4831a294> < 0, 0>
  pcn_output                             <54> <0xfd054c48, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
Decap:
  pcn_input                             <55> <0xfd054c48, 0x4830ba8c> < 0, 0>
  q_fq_input                             <96> <0xfd05f330, 0x48312c7c> < 0, 0>
  ipv4                                   <26> <0xfcc03e80, 0x00000000> < 0, 0>
Fixup:
  l2_adj_rewrite                         <86> <0xfcaa945c, 0x00000000> < 0, 0>
  pcn_output                             <54> <0xfd054c48, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
<13> (mpls)   Stats IN: 204 pkts, 23256 bytes; OUT: 0 pkts, 0 bytes
Encap:
  mpls                                   <25> <0xfcc7ddbc, 0x00000000> < 0, 0>
  ether                                   <30> <0xfd0189b4, 0x482c736c> < 0, 0>
  l2_adj_rewrite                         <86> <0xfcaa997c, 0x4831a2e8> < 0, 0>
  pcn_output                             <54> <0xfd0561f0, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
Decap:
  pcn_input                             <55> <0xfd0561f0, 0x4830ba8c> < 0, 0>
  q_fq_input                             <96> <0xfd05f330, 0x48312c7c> < 0, 0>
  mpls                                   <25> <0xfcc7b2b8, 0x00000000> < 152, 17328>
Fixup:
  l2_adj_rewrite                         <86> <0xfcaa945c, 0x00000000> < 0, 0>
  pcn_output                             <54> <0xfd0561f0, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
<22> (ether_sock) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
Encap:
  ether_sock                             <98> <0xfd01a774, 0x482c736c> < 0, 0>
  l2_adj_rewrite                         <86> <0xfcaa997c, 0x482d85f0> < 0, 0>
  pcn_output                             <54> <0xfd054bfc, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>
Decap:
  pcn_input                             <55> <0xfd054bfc, 0x4830ba8c> < 0, 0>
  q_fq_input                             <96> <0xfd05f330, 0x48312c7c> < 0, 0>
  ether_sock                             <98> <0xfd01a91c, 0x482c736c> < 0, 0>
Fixup:
  l2_adj_rewrite                         <86> <0xfcaa945c, 0x00000000> < 0, 0>
  pcn_output                             <54> <0xfd054bfc, 0x48319f04> < 0, 0>
  q_fq                                   <43> <0xfd05f4b8, 0x48320fec> < 0, 0>
  txm_nopull                             <60> <0xfcadba38, 0x4824c0fc> < 0, 0>

```

Protocol SAFI counts:

```

-----
      Protocol      SAFI      Pkts In      Bytes In      Pkts Out      Bytes Out
-----
      ipv4          Unicast          0          0          0          0

```

ipv4	Multicast	0	0	7	434
ipv4	Broadcast	0	0	0	0
ipv6	Unicast	0	0	0	0
ipv6	Multicast	0	0	0	0

The following fixed physical layer interface module (PLIM) example shows the ASIC counters. The output displays any drop counters or error counters incrementing for the interface.

```
RP/0/RP0/CPU0:router# show controllers plim asic statistics interface tenGigE 1/3/0/0
```

Node: 1/3/CPU0

-----  
TenGigE1/3/0/0 Tx Statistics  
-----

TotalOctets	: 195723291	TotalPkts	: 217711
UnicastPkts	: 74830	MulticastPkts	: 142879
BroadcastPkts	: 2	<64Octets	: 0
64Octets	: 5337	65to127Octets	: 73730
128to255Octets	: 1813	256to511Octets	: 2800
512to1023Octets	: 2846	1024to1518Octets	: 131185
1519to1548Octets	: 0	1549to9216Octets	: 0
>9216Octets	: 0	BadCRCPkts	: 0
802.1QPkts	: 33396	Underrun	: 0
Runt	: 0	Giant	: 0
PausePkts	: 0	Jabbers	: 0
DeferralAbort	: 0	LateCollision	: 0
CollisionAbort	: 0	OneCollision	: 0
MultiCollision	: 0	TotalCollisions	: 0
TotalDefer	: 0	LateCollisionAbort	: 0
LengthAbort	: 0	TxBP count	: 0

-----  
TenGigE1/3/0/0 Rx Statistics  
-----

TotalOctets	: 46777667	UnicastPkts	: 48672
TotalPkts	: 145028	BroadcastPkts	: 2
MulticastPkts	: 96354	65to127Octets	: 78648
64Octets	: 27411	256to511Octets	: 464
128to255Octets	: 15601	1024to1518Octets	: 22371
512to1023Octets	: 533	1549to9216Octets	: 0
1519to1548Octets	: 0	BadCRCPkts	: 0
>9216Octets	: 0	Runt	: 0
BadCodedPkts	: 0	802.1QPkts	: 33392
ShortPkts	: 0	PausePkts	: 0
Drop	: 0	Jabbers	: 0
ControlPkts	: 0		
BadPreamble	: 0		

-----  
TenGigE1/3/0/0 Drop  
-----

RxFIFO Drop	: 0	PAR Tail Drop	: 0
PAR Err Drop	: 0	MAC/VLAN Drop	: 33392
TxFIFO Drop	: 0		

```
RP/0/RP0/CPU0:router# show controllers plim asic statistics interface pos 0/5/0/0
```

Node: 0/5/CPU0

-----  
POS0/5/0/0 Tx Statistics  
-----

TotalOctets	: 85578609	TotalPkts	: 265727
UnicastPkts	: 265727	MulticastPkts	: 0
BroadcastPkts	: 0	<64Octets	: 70059
64Octets	: 0	65to127Octets	: 145269

```

128to255Octets      : 920
512to1023Octets     : 659
1519to1548Octets    : 1
>9216Octet          : 0
802.1QPkts          : 0
Runt                 : 0
PausePkts           : 0
DeferralAbort        : 0
CollisionAbort       : 0
MultiCollision       : 0
TotalDefer           : 0
LengthAbort          : 0
256to511Octets      : 126
1024to1518Octets    : 48622
1549to9216Octets    : 71
BadCRCPkts          : 0
Underrun             : 0
Giant                : 0
Jabbers              : 0
LateCollision        : 0
OneCollision         : 0
TotalCollisions      : 0
LateCollisionAbort   : 0
TxBP count           : 0

```

#### POS0/5/0/0 Rx Statistics

```

-----
TotalOctets          : 67976561
TotalPkts            : 159586
MulticastPkts        : 0
64Octets             : 0
128to255Octets       : 1534
512to1023Octets      : 782
1519to1548Octets     : 0
>9216Octets          : 0
BadCodedPkts         : 0
ShortPkts             : 63514
Drop                 : 0
ControlPkts          : 0
BadPreamble          : 0
UnicastPkts          : 159586
BroadcastPkts        : 0
65to127Octets        : 51860
256to511Octets       : 555
1024to1518Octets     : 41278
1549to9216Octets     : 63
BadCRCPkts           : 0
Runt                  : 0
802.1QPkts           : 0
PausePkts            : 0
Jabbers              : 0

```

#### POS0/5/0/0 Drop

```

-----
RxFIFO Drop          : 1761
PAR Tail Drop        : 0
TxFIFO Drop          : 0

```

The following 1 Gigabit Ethernet example shows the shared port adapter (SPA) counters. The output displays any drop counters or error counters incrementing for the interface.

```
RP/0/RP0/CPU0:router# show hw-module subslot 0/0/0 counters mac 0
```

```

SPA 0/0/0 device mac 0/0 info:
port:0
good_octets_received: 87538984
bad_octets_received: 41593
good_frames_received: 120401
bad_frames_received: 71
broadcast_frames_received: 1
multicast_frames_received: 115293
good_octets_sent: 54953513
good_frames_sent: 104469
broadcast_frames_sent: 0
multicast_frames_sent: 0
mac_transfer_error: 0
excessive_collision: 0
unrecog_mac_control_received: 0
fc_sent: 0
good_fc_received: 0
rx_over_flow_events: 0
undersize: 0
fragments: 0
oversize: 0
jabber: 0
mac_rcv_error: 71
bad_crc: 0
collisions: 0

```

```
late_collision: 0
rate_limit_dropped: 0
spi4_rx_frames: 0
spi4_tx_frames: 0
```

The following Packet-over-SONET/SDH (POS) port example shows the SPA counters. The output displays any drop counters or error counters incrementing for the interface.

```
RP/0/RP0/CPU0:router# show hw-module subslot counters framer
```

```
SPA 0/0/1 device framer 0/0 info:
```

```
Egress:
```

	Pkts	Bytes	Underruns	Aborts
port 0	90027	213674250	0	0
port 1	0	0	0	0
port 2	0	0	0	0
port 3	0	0	0	0

```
Ingress:
```

	Pkts	Bytes	CRC errs	Runts	Giants	Aborts
port 0	200042	222631936	0	0	0	0
port 1	0	0	0	0	0	0
port 2	0	0	0	0	0	0
port 3	0	0	0	0	0	0

```
SPA 0/0/5 device framer 0/0 info:
```

```
Port_1 (port 0) framer registers:
```

```
Port_1 Framer counters:
```

```
STREAM 0
```

```

Rx Bytes (48-bit) (#0x60286078-0x883c): 3951749
Rx Good Bytes (48-bit) (#0x60286080-0x8840): 3112256
Rx Good Packets (48-bit) (#0x60286040-0x8820): 194516
Tx Byte Cnt Reg (48-bit) (#0x6028a070-0xa838): 3905343
Tx Good Bytes Cnt Reg (48-bit) (#0x6028a068-0xa834): 3112256
Tx Transmitted Packet Cnt Reg (48-bit) (#0x6028a040-0xa820): 194516
```

## Verifying and Troubleshooting Pluggable Optical Interfaces

Troubleshooting the pluggable optical interfaces includes verifying that you have an optical card installed, enabled, and functioning properly. To troubleshoot the configured pluggable optical interfaces, perform the following procedure.

### SUMMARY STEPS

1. **show hw-module subslot *address* brief pluggable-optics**
2. **show hw-module subslot *address* status pluggable-optics**
3. **show hw-module subslot *address* errors pluggable-optics**
4. **show hw-module subslot *address* registers pluggable-optics**
5. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show hw-module subslot address brief pluggable-optics</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show hw-module subslot 0/6/4 brief pluggable-optics	Displays a brief summary of the pluggable optics status for all nodes, including optics type, vendor, and state. Check that the state is enabled for the node that you are troubleshooting.
Step 2	<b>show hw-module subslot address status pluggable-optics</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show hw-module subslot 0/6/4 status pluggable-optics	Displays the status of the pluggable optics for the specified node, including faults and environmental data. Check that the state and the transceiver are enabled. Check for any warnings or alarms.
Step 3	<b>show hw-module subslot address errors pluggable-optics</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show hw-module subslot 0/6/4 errors pluggable-optics	Displays any errors that are present on the node. Note if there are any errors.  Verify that Phased Initialization displays Phase Reached: 4. Verify that Socket Verification displays “passed” for both Compatibility and Security.
Step 4	<b>show hw-module subslot address registers pluggable-optics</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show hw-module subslot 0/6/4 registers pluggable-optics	Displays all available information on the optics including the IDROM contents.
Step 5	Contact Cisco Technical Support.	For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

The following example shows typical outputs for these commands on systems.

```
RP/0/RP0/CPU0:router#show hw-module subslot 0/6/4 brief pluggable-optics
```

```
SPA 0/6/4 device pluggable-optics 0/0 info:
SPA 0/6/4 device pluggable-optics 1/0 info:
SPA 0/6/4 device pluggable-optics 2/0 info:
SPA 0/6/4 device pluggable-optics 3/0 info:
SPA 0/6/4 device pluggable-optics 4/0 info:
int sonet 6/4/4:
    ID: SFP
    Extended ID: 4
    Xcvr Type: OC12 SR-1/STM4 MM (8)
    Connector: LC
    Vendor name: CISCO-OCP
    Vendor part number: TRPD12MM2EAS-CS
    State: Enabled

SPA 0/6/4 device pluggable-optics 5/0 info:

int sonet 6/4/5:
    ID: SFP
    Extended ID: 4
```

```
Xcvr Type: OC12 SR-1/STM4 MM (8)
Connector: LC
Vendor name: CISCO-OCF
Vendor part number: TRPD12MM2EAS-CS
State: Enabled

SPA 0/6/4 device pluggable-optics 6/0 info:

int sonet 6/4/6:
  ID: SFP
  Extended ID: 4
  Xcvr Type: OC12 SR-1/STM4 MM (8)
  Connector: LC
  Vendor name: CISCO-OCF
  Vendor part number: TRPD12MM2EAS-CS
  State: Enabled

SPA 0/6/4 device pluggable-optics 7/0 info:

RP/0/RP0/CPU0:router#show hw-module subslot 0/6/4 status pluggable-optics

SPA 0/6/4 device pluggable-optics 0/0 info:
SPA 0/6/4 device pluggable-optics 1/0 info:
SPA 0/6/4 device pluggable-optics 2/0 info:
SPA 0/6/4 device pluggable-optics 3/0 info:
SPA 0/6/4 device pluggable-optics 4/0 info:
int sonet 6/4/4:
  State: Enabled
  Environmental Information - raw values
    Temperature: 42.57 C
    Supply voltage: 32669 in units of 100uVolt
    Tx bias: 0 in units of 2uAmp
    Tx power: -40 dBm (0 in units of 0.1 uW)
    Rx power: -16 dBm (182 in units of 0.1 uW)

  Transceiver: Enabled
  SW TX Fault: unavailable
  SW LOS: unavailable
  No active alarms
  No active warnings
  Version Identifier (VID): V01
  Product Identifier (PID): SFP-OC12-MM
  Part Number (PN): 10-2079-01
  CLEI: IPUAG9RAA

SPA 0/6/4 device pluggable-optics 5/0 info:
int sonet 6/4/5:
  State: Enabled
  Environmental Information - raw values
    Temperature: 41.207 C
    Supply voltage: 32816 in units of 100uVolt
    Tx bias: 0 in units of 2uAmp
    Tx power: -40 dBm (0 in units of 0.1 uW)
    Rx power: -16 dBm (194 in units of 0.1 uW)

  Transceiver: Enabled
  SW TX Fault: unavailable
  SW LOS: unavailable
  No active alarms
  No active warnings
  Version Identifier (VID): V01
  Product Identifier (PID): SFP-OC12-MM
  Part Number (PN): 10-2079-01
```

```

CLEI: IPUIAG9RAA

SPA 0/6/4 device pluggable-optics 6/0 info:
int sonet 6/4/6:
    State: Enabled
    Environmental Information - raw values
        Temperature: 41.47 C
        Supply voltage: 32816 in units of 100uVolt
        Tx bias: 0 in units of 2uAmp
        Tx power: -40 dBm (0 in units of 0.1 uW)
        Rx power: -18 dBm (142 in units of 0.1 uW)

    Transceiver: Enabled
    SW TX Fault: unavailable
    SW LOS: unavailable
    No active alarms
    No active warnings
    Version Identifier (VID): V01
    Product Identifier (PID): SFP-OC12-MM
    Part Number (PN): 10-2079-01
    CLEI: IPUIAG9RAA

SPA 0/6/4 device pluggable-optics 7/0 info:

RP/0/RP0/CPU0:Pl_CRS-8#
RP/0/RP0/CPU0:router#show hw-module subslot 0/6/4 errors pluggable-optics

SPA 0/6/4 device pluggable-optics 0/0 info:
SPA 0/6/4 device pluggable-optics 1/0 info:
SPA 0/6/4 device pluggable-optics 2/0 info:
SPA 0/6/4 device pluggable-optics 3/0 info:
SPA 0/6/4 device pluggable-optics 4/0 info:

int sonet 6/4/4:
Phased Initialization
    Phase Reached: 4
    Phase Exit Code: Success 0
    Phase Read Offset: 256

Socket Verification
    Compatibility: Compatibility passed
    Security: Security passed

SPA 0/6/4 device pluggable-optics 5/0 info:
int sonet 6/4/5:
Phased Initialization
    Phase Reached: 4
    Phase Exit Code: Success 0
    Phase Read Offset: 256

Socket Verification
    Compatibility: Compatibility passed
    Security: Security passed

SPA 0/6/4 device pluggable-optics 6/0 info:
int sonet 6/4/6:
Phased Initialization
    Phase Reached: 4
    Phase Exit Code: Success 0
    Phase Read Offset: 256

Socket Verification
    Compatibility: Compatibility passed

```



Security: Security passed

SPA 0/6/4 device pluggable-optics 7/0 info:

RP/0/RP0/CPU0:router#show hw-module subslot 0/6/4 registers pluggable-optics

Tue Aug 17 05:29:36.306 DST

SPA 0/6/4 device pluggable-optics 0/0 info:

SPA 0/6/4 device pluggable-optics 1/0 info:

SPA 0/6/4 device pluggable-optics 2/0 info:

SPA 0/6/4 device pluggable-optics 3/0 info:

SPA 0/6/4 device pluggable-optics 4/0 info:

int sonet 6/4/4:

ID: SFP

Extended ID: 4

Xcvr Type: OC12 SR-1/STM4 MM (8)

Connector: LC

Encoding: reserved

Bit Rate: 600 Mbps

50 micron-multimode fiber supported length: 500 m

62.5 micron-multimode fiber supported length: 500 m

Upper bit rate limit: not specified

Lower bit rate limit: not specified

Date code (yy/mm/dd): 05/06/17

Vendor name: CISCO-OCF

Vendor OUI: 2589

Vendor Part Number (PN): TRPD12MM2EAS-CS

Vendor Rev: A

Vendor SN (SN): OCP09210623

Options implemented:

LOS Signal

TX Disable Signal

Enhanced options implemented:

Alarm/Warning Flags

Diagnostic monitoring implemented:

Internally Calibrated

Digital Diagnostic Monitoring

Idprom contents (hex):

0x00: 03 04 07 00 00 10 00 00 00 00 00 05 06 00 00 00

0x10: 32 32 00 00 43 49 53 43 4F 2D 4F 43 50 20 20 20

0x20: 20 20 20 20 00 00 0A 1D 54 52 50 44 31 32 4D 4D

0x30: 32 45 41 53 2D 43 53 20 41 20 20 20 05 1E 00 FD

0x40: 00 12 00 00 4F 43 50 30 39 32 31 30 36 32 33 20

0x50: 20 20 20 20 30 35 30 36 31 37 20 20 68 80 01 87

0x60: 00 00 03 B5 E1 1E 6C A6 32 1B 46 E4 18 30 17 DE

0x70: 2A 95 A1 00 00 00 00 00 00 00 00 00 9C 5F 07 F7

0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Status/Control Register: 0000

Alarm Status: 0000

Warning Status: 0000

#### THRESHOLDS

		high alarm	high warning	low warning	low alarm
Temperature	C	+098.000	+095.000	-02.000	-03.000
Voltage	V	003.5100	003.4950	003.1150	003.1000
Bias Current	mA	000.0000	000.0000	000.0000	000.0000
Transmit power	mW	000.0000	000.0000	000.0000	000.0000
Receive power	mW	000.0631	000.0398	000.0008	000.0005

Diagnostics contents (hex):

0x00: 62 00 FD 00 5F 00 FE 00 89 1C 79 18 88 86 79 AE

0x10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0x20: 02 77 00 05 01 8E 00 08 00 00 00 00 00 00 00

```

0x30:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40:  00 00 00 00 3F 80 00 00 00 00 00 00 01 00 00 00
0x50:  01 00 00 00 01 00 00 00 01 00 00 00 00 00 00 FF
0x60:  22 21 80 30 00 00 00 00 00 00 AB 00 00 00 00 00
0x70:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x80:  49 50 55 49 41 47 39 52 41 41 31 30 2D 32 30 37
0x90:  39 2D 30 31 56 30 31 20 8A FB 55 00 00 00 00 6B
0xA0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xB0:  00 00 00 00 25 A8 00 00 00 00 00 00 CD 00 00 AA AA
0xC0:  53 46 50 2D 4F 43 31 32 2D 4D 4D 20 20 20 20 20
0xD0:  20 20 20 20 00 00 00 00 00 00 00 00 00 00 00 F2
0xE0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xF0:  00 00 00 00 00 00 00 00 00 00 40 00 40 00 00 00

```

--More--

## Troubleshooting Common Issues with Bundle Interfaces

Bundle interfaces utilize features and services provided by different components. To troubleshoot the common issues with bundle interface components on Cisco IOS XR software, perform the following procedure.

### SUMMARY STEPS

1. **show tech-support bundles file** *filename*
2. **show bundle** *bunde-name*
3. **show iir interfaces name** *bundle-name*
4. **show iir interfaces name** *bundle-name* **location** *location-id*
5. **show pfi-ifh database** *bundle-name* **location** *location-id*
6. **show cef adjacency** *bundle-name* **hardware egress detail** *location* *location-id*
7. **show cef adjacency** *bundle-name* **hardware ingress remote detail** *location* *location-id*
8. Contact Cisco Technical Support if the problem is not resolved.

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show tech-support bundles file</b> <i>filename</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show tech-support bundles file filename	Collects the output from bundle in both normal and admin modes and stores it in a file. It is important to do this as soon as possible after observing a problem because the traces with valuable information for debugging may wrap and be lost. It is strongly recommended that the command output be sent directly to file, rather than screen scraped, to avoid loss/corruption of part of the logs.
Step 2	<b>show bundle</b> <i>bunde-name</i>  <b>Example:</b> RP/0/RP0/CPU0:router# sh bundle bundle-ether 202	Make sure that the member links are in Distributing (4) state.

	Command or Action	Purpose
Step 3	<b>show iir interfaces name</b> <i>bundle-name</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show iir interfaces name bundle-ether 202	Ensure that the member links are in ACTIVE state.
Step 4	<b>show iir interfaces name</b> <i>bundle-name</i> <b>location</b> <i>location-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show iir interfaces name bundle-ether 202 location 0/RP0/CPU0	Make sure the IIR database of Bundle Interface Information is in sync.
Step 5	<b>show pfi-ifh database</b> <i>bundle-name</i> <b>location</b> <i>location-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show pfi-ifh database bundle-ether 205 location 0/1/CPU0	Make sure that the PFI-IFH database of bundle interface info is in sync.
Step 6	<b>show cef adjacency</b> <i>bundle-name</i> <b>hardware egress detail</b> <i>location</i> <i>location-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef adjacency bundle-name hardware egress detail location 0/RP0/CPU0 RP/0/RP0/CPU0:router# show cef adjacency bundle-name hardware ingress remote detail location 0/RP0/CPU0	Make sure that the TLU entries are allocated and bundle adjacency info is properly programmed
Step 7	Contact Cisco Technical Support.	For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.





## CHAPTER 6

# Troubleshooting the Control Plane Ethernet Network

---

This chapter describes techniques that you can use to troubleshoot the control plane Ethernet network on routers using Cisco IOS XR software. It includes the following sections:

- [Cisco CRS-1 Control Plane Ethernet Network Overview, page 6-147](#)
- [Using the Online Diagnostics Tools, page 6-148](#)
- [Troubleshooting Booting the System Control Plane Ethernet Network, page 6-149](#)
- [Troubleshooting the Multishelf System Router Topology, page 6-153](#)
- [Troubleshooting the CRS-1, 4-slot, 8-slot, or 16-slot System Router Topology, page 6-159](#)

## Cisco CRS-1 Control Plane Ethernet Network Overview

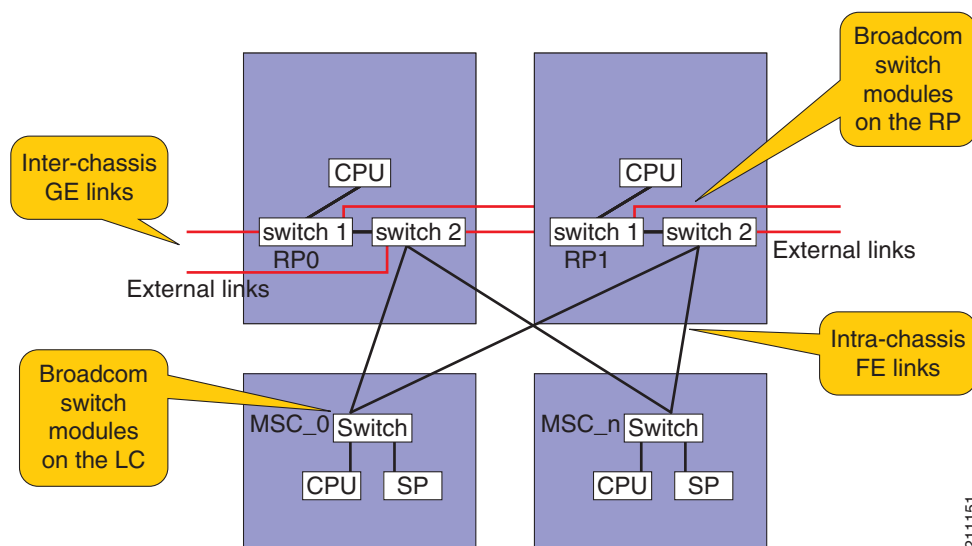
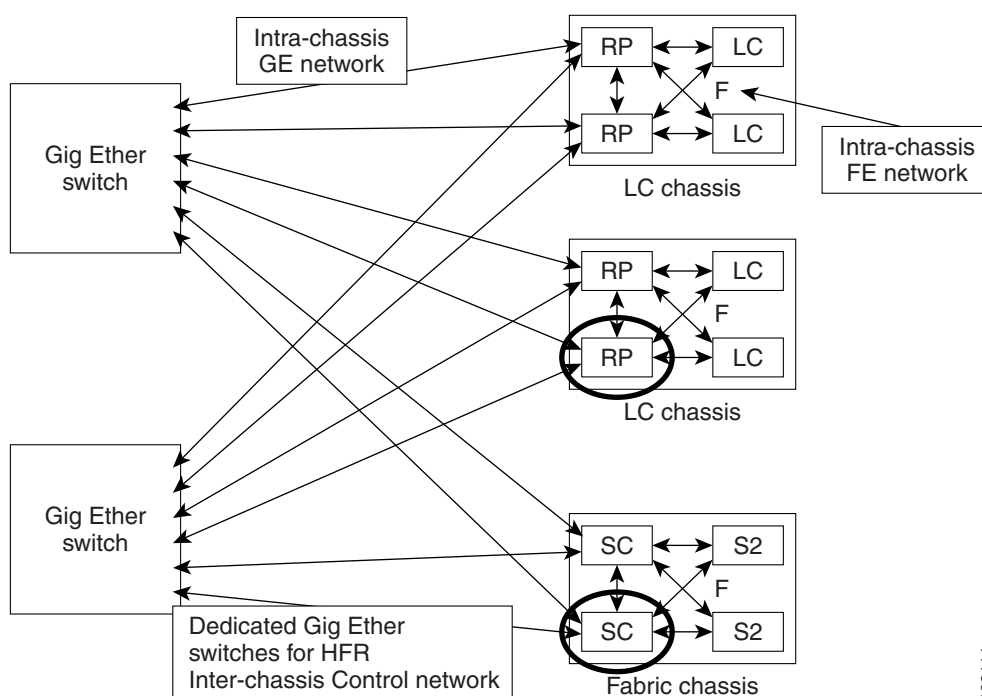
The system control plane Ethernet network is used for processes on different devices to communicate for functions such as system device discovery, image transfers, heartbeat messages, alarms, and configuration management.

All devices in a system using Cisco IOS XR software connect to the system control plane Ethernet network. There are two control planes in the control Ethernet topology:

- **Intrachassis**—Within a chassis, the control plane is provided using Fast Ethernet (FE) links between nodes. The FE links are internal to the chassis and cannot be removed.
- **Interchassis**—System Control Plane connectivity between chassis in the system is provided using Gigabit Ethernet (GE) links. Two links are presented on the front of each of the route processors (active and standby) in each of the line card chassis (LCCs) and shelf controllers (active and standby) in the fabric card chassis (FCC). Two ports are provided for redundancy purposes. This control plane is supported on the Cisco CRS-1 Multishelf System only.

See [Figure 6-1](#) for an illustration of the Cisco CRS-1 control plane Ethernet network

See [Figure 6-2](#) for an illustration of the Cisco CRS-1 Multishelf System control plane Ethernet network. Two external Catalyst switches (WS-C6509-NEB-A) are used to provide connectivity between the FCC and LCCs. In the Cisco CRS-1 Multishelf System, the system control plane links from each route processor (RP) and shelf controller (SC) connect to an external Catalyst switch. RPs and SCs are not directly connected to each other. See [Figure 6-2](#).

**Figure 6-1 Cisco CRS-1 Control Ethernet Topology****Figure 6-2 Cisco CRS-1 Multishelf System Control Ethernet Topology**

## Using the Online Diagnostics Tools

The online diagnostics tools can alert you to potential problems in the control plane connections between the Fabric Card Chassis and the Line Card Chassis. For information on using the online diagnostics, see the [“Using Diagnostic Commands”](#) section on page 7-166.

# Troubleshooting Booting the System Control Plane Ethernet Network

To verify and troubleshoot booting the Cisco CRS-1 system control plane Ethernet network, perform the following procedures.

## SUMMARY STEPS

1. **show platform**
2. **admin debug shelfmgr boot**
3. **show controllers backplane ethernet clients all location** *node-id*
4. **show controllers backplane ethernet clients 13 statistics location** *node-id*
5. **show controllers switch {0 | 1} statistics location** *node-id*
6. Place the designated shelf controller (DSC) in ROM Monitor (ROMMON) mode:
  - a. **admin**
  - b. **config register 0x0**
  - c. **exit**
  - d. **reload**
7. **show_bcm_links** (Cisco CRS-1 Multishelf System only)
8. Exit ROMMON mode:
  - a. **confreg 0x102**
  - b. **reset**
9. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show platform</b>	Displays information about the status of cards and modules installed in the router.
	<b>Example:</b> RP/0/RP0/CPU0:router# show platform	Verify that the expected nodes display IOS XR RUN under the State column of the command output.
Step 2	<b>admin debug shelfmgr boot</b>	If no output is displayed, the shelfmgr is not receiving boot requests.
	<b>Example:</b> RP/0/RP0/CPU0:router# admin debug shelfmgr boot	

	Command or Action	Purpose
Step 3	<p><b>show controllers backplane ethernet clients all location node-id</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show controllers backplane ethernet clients all location 0/RP1/CPU0</pre>	<p>Displays information about all local client applications. Each row contains the client ID, product identifier (PID) of the process that is registered to receive packets with that client ID and a description of the client.</p> <p>The eth_server allows client processes to send and receive packets over the control Ethernet. eth_server uses client IDs to demultiplex packets that arrive at the node.</p> <p>Two client IDs in the output are important for troubleshooting boot problems:</p> <ul style="list-style-type: none"> <li>Client Ethernet server ID 13—used for boot requests</li> <li>Client Ethernet server ID 4—used for heartbeats</li> </ul>
Step 4	<p><b>show controllers backplane ethernet local clients 13 statistics location node-id</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show controllers backplane ethernet clients 13 statistics location 0/RP1/CPU0</pre>	<p>Displays a list of client statistics for the specified client ID. Check the values for:</p> <ul style="list-style-type: none"> <li>Packets input</li> <li>Packets delivered</li> </ul> <p>If they contain values other than 0, boot requests have been received and replies have been sent (packets output). Proceed to <a href="#">Step 5</a>.</p> <p>If they contain values of 0, check the system control plane Ethernet network physical connectivity. See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on system control plane Ethernet network cabling.</p> <p>If there are no problems with the physical connectivity, contact Cisco Technical Support. For Cisco Technical Support contact information, see the “<a href="#">Obtaining Documentation and Submitting a Service Request</a>” section on page viii in the <a href="#">Preface</a>.</p>
Step 5	<p><b>show controllers switch {0   1} statistics location node-id</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show controllers switch 0 statistics location 0/RP1/CPU0</pre>	<p>Display statistics on all ports on the switch controllers. The <b>location node-id</b> keyword and argument are required for obtaining information on the remote RPs and SCs.</p>
Step 6	<p><b>admin</b> <b>config-register 0x0</b> <b>exit</b> <b>reload</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# admin RP/0/RP0/CPU0:router(admin)# config-register 0x0 RP/0/RP0/CPU0:router(admin)# exit RP/0/RP0/CPU0:router# reload</pre>	<p>Places the DSC in ROMMON mode.</p>



	Command or Action	Purpose
<b>Step 7</b>	<b>show_bcm_links</b>  <b>Example:</b> rommon B1 > show_bcm_links	<p><b>Note</b> This step is supported on Cisco CRS-1 Multishelf Systems only.</p> <p>This command is used to confirm connectivity from the RP or SC to the Catalyst switch in ROM monitor (ROMMON) mode. See <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for detailed information on entering ROMMON mode.</p> <p>Displays the connectivity state of the links to the onboard Ethernet switch as well as the system control plane links to the standby RP and remote Catalyst switch.</p> <p>Verify that all expected active ports are displayed. If no connectivity is detected by a particular port, it is not displayed in the output. GE Port 1 must be active, because this is the port on the switch used for external connectivity. GE Port 2 is the link to the secondary switch, so it should also be active.</p>
<b>Step 8</b>	<b>confreg 0x102</b> <b>reset</b>  <b>Example:</b> rommon B2 > confreg 0x102 rommon B3 > reset	Exits ROMMON mode, resetting and initializing the router.
<b>Step 9</b>	Contact Cisco Technical Support if the problem is not resolved.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The following example shows that all expected nodes in the Cisco CRS-1 Multishelf System are in the 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	40C192-POS/DPT	IOS XR RUN	PWR, NSHUT, MON
0/8/SP	MSC (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/8/CPU0	MSC	160C48-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/FC0/SP	LCC-FAN-CT (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/FC1/SP	LCC-FAN-CT (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/AM1/SP	ALARM (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM0/SP	FC/M (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM1/SP	FC/M (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM4/SP	FC/M (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM5/SP	FC/M (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/RP0/CPU0	RP (Active)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/RP1/CPU0	RP (Standby)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/FC0/SP	LCC-FAN-CT (SP)	N/A	IOS XR RUN	PWR, NSHUT, MON

1/FC1/SP	LCC-FAN-CT(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/AM1/SP	ALARM(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/SM0/SP	FC/M(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/SM1/SP	FC/M(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/SM4/SP	FC/M(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
1/SM5/SP	FC/M(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
F0/SM0/SP	FCC-SFC(SP)	FCC-FM-1S	IOS XR RUN	PWR, NSHUT, MON
F0/SM1/SP	FCC-SFC(SP)	FCC-FM-1S	IOS XR RUN	PWR, NSHUT, MON
F0/SM4/SP	FCC-SFC(SP)	FCC-FM-1S	IOS XR RUN	PWR, NSHUT, MON
F0/SM5/SP	FCC-SFC(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
F0/SC0/CPU0	FCC-SC(Standby)	N/A	IOS XR RUN	PWR, NSHUT, MON
F0/SC1/CPU0	FCC-SC(Active)	N/A	IOS XR RUN	PWR, NSHUT, MON
F0/AM1/SP	ALARM(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
F0/LM0/SP	FCC-LED(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON

The following example shows the current state of each eth_server client in the Cisco CRS-1 Multishelf System:

```
RP/0/RP0/CPU0:router# show controllers backplane ethernet clients all
```

Intf Name	Client ethernet server id	Client Process Id	Description
-----			
FE0_RP1_0	1	24601	QNX network manager
	2	53317	Group services
	3	0	Reserved for Attach
	4	53338	Plugin controller
	5	53298	Designated SC
	...	...	...
	13	53338	Card Configuration Protocol
	...	...	...
	22	0	Test client out-of-band

The following example shows that there are 18 nodes in the Cisco CRS-1 Multishelf System in the run state, which means that 18 boot requests have been received by eth_server and 18 replies have been sent:

```
RP/0/RP0/CPU0:router# show controller backplane ethernet clients 13 statistics location 0/RP1/CPU0
```

```
Client ShelfMgr, ES Client Id 13, PID 53338 running on FastEthernet0_RP1_0
 18 packets input, 8676 bytes
 18 packets delivered, 8676 bytes
 0 packets discarded (0 bytes) in garbage collection
 0 (0 bytes) unicast packets filtered
 0 (0 bytes) multicast packets filtered
 0 (0 bytes) buffer mgmt policy discards
 0 (0 bytes) locking error discards
 18 packets output, 8676 bytes, 0 could not be transmitted
```

The following example shows that the switch located on the RP sent data towards Port 2 that connects to line card 1 (LC1) in the Cisco CRS-1 Multishelf System:

```
RP/0/RP0/CPU0:router# show controllers switch 1 statistics
```

Port	Tx Frames	Tx Errors	Rx Frames	Rx Errors	Connects
-----					
1 :	382714	0	132876	2	0/LC0
2 :	0	0	0	0	0/LC1

The following example shows how to place the DSC in ROMMON mode:

```
RP/0/RP0/CPU0:router# admin
RP/0/RP0/CPU0:router(admin)# config-register 0x0
```

```
Successfully set config-register to 0x0 on node 0/0/CPU0
```

```
RP/0/RP0/CPU0:router(admin)# exit
RP/0/RP0/CPU0:router# reload
```

```
Proceed with reload? [confirm]
System Bootstrap, Version 12.0(20040624:164256) [assafb-misc1 1.14dev(0.91)] DEV
ELOPMENT SOFTWARE
Copyright (c) 1994-2004 by cisco Systems, Inc.
DRAM DIMM Slot 1: 512M found, Slot 2: Empty
MPC7450 platform with 524288 Kbytes of main memory
rommon 1 >
```

The **show_bcm_links** command, run in ROMMON mode, shows that all expected links are active on the Cisco CRS-1 Multishelf System:

```
rommon B1 > show_bcm_links
```

```
Ports Active on Switch 0
FE Port 1
GE Port 1
GE Port 2
Ports Active on Switch 1
GE Port 2
```

The following example shows how to exit ROMMON mode:

```
rommon B1> confreg
```

```
Configuration Summary
(Virtual Configuration Register: 0x0)
enabled are:
console baud: 9600
boot: the ROM Monitor
do you wish to change the configuration? y/n [n]: y
enable "diagnostic mode"? y/n [n]: n
change console baud rate? y/n [n]: n
change the boot characteristics? y/n [n]: y
enter to boot:
0 = ROM Monitor
1 = MBI Validation Boot Mode
[0]: 1
Configuration Summary
(Virtual Configuration Register: 0x102)
enabled are:
console baud: 9600
boot: image specified by the boot system commands
do you wish to change the configuration? y/n [n]: n
You must reset or power cycle for new config to take effect

rommon B2> reset
```

## Troubleshooting the Multishelf System Router Topology

To verify and troubleshoot the Cisco CRS-1 Multishelf System router topology, perform the following procedures.

Each SC and RP has two internal switches (0 and 1). Each internal switch has two Gigabit Ethernet control ports. Port 1 is on the front panel and is used as a control port for interchassis connectivity (Gigabit Ethernet control). Port 2 is used to connect between the two internal switches in the SC. The Fast Ethernet (FE) ports of the switches are used within the chassis to connect to other nodes (intrachassis connections).

If an external Catalyst 6000 Series Switches or Integrated Control Switches (SC-GE-22) are used, use the **show spantree mst 1** or **show udld gigabit ethernet** command to show the ports.

## SUMMARY STEPS

1. **show platform**
2. Verify that all physical Catalyst switch connections are made.
3. **show controllers switch 0 ports location *node-id***
4. **show controllers switch 1 ports location *node-id***
5. **show spantree mst 1 brief location *node-id***
6. **show controllers switch 0 statistics location *node-id***
7. **show controllers switch 1 statistics location *node-id***
8. **show controllers backplane ethernet detail location *node-id***
9. Place the DSC in ROMMON mode:
  - a. **admin**
  - b. **config register 0x0**
  - c. **exit**
  - d. **reload**
10. **show_bcm_links**
11. Exit ROMMON mode:
  - a. **confreg 0x102**
  - b. **reset**
12. Contact Cisco Technical Support if the problem is not resolved

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show platform</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show platform	Displays information and status on each node in the system.  Verify that all components in the router are visible and that the components are in the run state.  If components are not visible in the command output, proceed to <a href="#">Step 3</a> and check the GE switch connections as illustrated in <a href="#">Figure 6-2</a> .
Step 2	Check that all external Catalyst switches are connected to the RPs and SCs as illustrated in <a href="#">Figure 6-2</a> .	Ensures that the connections for the control plane Ethernet network are made. See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on system control plane Ethernet network to cabling.
Step 3	<b>show controllers switch 0 ports location node-id</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 0 ports location 0/RP1/CPU0	Displays status on a Switch Port 0.  Verify that all ports are in the forwarding state.
Step 4	<b>show controllers switch 1 ports location node-id</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 1 ports location 0/RP1/CPU0	Displays status on a switch port 1.  Verify that all ports are in the forwarding state.
Step 5	<b>show spantree mst 1 brief location node-id</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show spantree mst 1 brief location 0/RP1/CPU0	Displays the spanning tree port details.  Verify that the designated root MAC address is the MAC address of the external Catalyst switch. Use the <b>show running-config</b> command to display the MAC address of the external Catalyst switch.
Step 6	<b>show controllers switch 0 statistics location node-id</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 0 statistics location 0/RP1/CPU0	Displays statistics on Switch Port 0.  Verify that the expected internal switch ports are carrying traffic (Tx and Rx frames).
Step 7	<b>show controllers switch 1 statistics location node-id</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 1 statistics location 0/RP1/CPU0	Displays statistics on Switch Port 1.  Verify that the expected internal switch ports are carrying traffic (Tx and Rx frames).

	Command or Action	Purpose
Step 8	<b>show controllers backplane ethernet detail</b> <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers backplane ethernet detail location 0/RP1/CPU0	Displays detailed information about backplane interfaces in a specific location.
Step 9	<b>admin</b> <b>config-register 0x0</b> <b>exit</b> <b>reload</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin RP/0/RP0/CPU0:router(admin)# config-register 0x0 RP/0/RP0/CPU0:router(admin)# exit RP/0/RP0/CPU0:router# reload	Places the DSC in ROMMON mode.
Step 10	<b>show_bcm_links</b>  <b>Example:</b> rommon B1 > show_bcm_links	<p>This command is used to confirm connectivity from the RP or SC to the Catalyst switch in ROM monitor (ROMMON) mode. See <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for information on entering ROMMON mode.</p> <p>Displays the connectivity state of the links to the onboard Ethernet switch as well as the system control Ethernet plane links to the standby RP and remote Catalyst switch.</p> <p>Verify that all expected active ports are displayed. If no connectivity is detected by a particular port, it will not be displayed in the output. GE Port 1 must be active as this is the port on the switch used for external connectivity. GE Port 2 is the link to the secondary switch, so it should also be active.</p>
Step 11	<b>confreg 0x102</b> <b>reset</b>  <b>Example:</b> rommon B2 > confreg 0x102 rommon B3 > reset	Exits ROMMON mode, resetting and initializing the router.
Step 12	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the <a href="#">Preface</a> .

### Examples

The output from the **show platform** command indicates that all expected nodes are in the IOS XR RUN state:

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

Node	Type	PLIM	State	Config State
-----				

0/0/SP	MSC(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/0/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/RP0/CPU0	RP(Standby)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/RP1/CPU0	RP(Active)	N/A	IOS XR RUN	PWR, NSHUT, MON
0/SM0/SP	FC/S(SP)	N/A	IOS XR RUN	PWR, NSHUT, MON

The following example shows that the appropriate ports on Switch Port 1 are forwarding:

```
RP/0/RP0/CPU0:router# show controllers switch 1 ports location 0/RP1/CPU0
```

```
Ports Active on Switch 1
FE Port 1 STP State : FORWARDING (Connected to - 0/RP0)
FE Port 2 STP State : FORWARDING (Connected to - 0/RP1)
FE Port 9 STP State : FORWARDING (Connected to - 0/SM0)
GE Port 1 STP State : FORWARDING // connected to External switch through the front
panel
GE Port 2 STP State : FORWARDING // connected to the other Internal switch
```

The following example displays the spanning tree details including the designated root (expected MAC address for the external Catalyst switch):

```
RP/0/RP0/CPU0:router# show spantree mst 1 brief location 0/rp1/cpu0
```

```
Instance 1
Vlans mapped: 1

Designated Root 00-0e-39-fe-70-00
Designated Root Priority 1 (0 + 1)
Designated Root Port GE_Port_0

Bridge ID MAC ADDR 00-05-9a-3e-89-55
Bridge ID Priority 32769 (32768 + 1)
Bridge Max Age 20 sec Hello Time 2 sec Forward Delay 15 sec Max Hops 4

Switched Interface State Role Cost Prio Type
-----
FE_Port_0 FWD desg 200000 128 P2P
GE_Port_0 FWD root 20000 128 P2P
GE_Port_1 DWN desg 20000 128 P2P
```

The following example shows the statistics for Switch Port 0:

```
RP/0/RP0/CPU0:router# show controllers switch 0 statistics location 0/RP1/CPU0
```

Port	Tx Frames	Tx Errors	Rx Frames	Rx Errors	Connects
1 :	0	0	0	0	0/LC0
2 :	0	0	164794	1	0/LC1
3 :	0	0	0	0	0/LC2
4 :	0	0	0	0	0/LC3
5 :	0	0	0	0	0/LC4
6 :	0	0	0	0	0/LC5
7 :	0	0	0	0	0/LC6
8 :	0	0	0	0	0/LC7
9 :	0	0	164799	1	0/LC8
10 :	0	0	0	0	0/LC9
11 :	0	0	0	0	0/LC10
12 :	0	0	0	0	0/LC11
13 :	0	0	0	0	0/LC12
14 :	0	0	0	0	0/LC13
15 :	0	0	0	0	0/LC14
16 :	0	0	0	0	0/LC15

```

25 :          0          0          0          0          GE_1
26 :          0          0      4370089          0      Switch 0

```

The following example shows output from the **show controller backplane ethernet location detail** command:

```
RP/0/RP0/CPU0:router# show controllers backplane ethernet detail location 0/1/0
```

```

FastEthernet0_1_0 is up
Hardware is 10/100 Ethernet, H/W address is 5246.4800.0010
Internet address is 10.0.0.16
MTU 1514 bytes
Encapsulation HFRIES (HFR Internal Ethernet Server)
Mode : Full Duplex, Rate : 100Mb/s
426422 packets input, 0 bytes, 1 total input drops
14170 packets discarded (935122 bytes) in garbage collection
16 packets discarded (5344 bytes) in recv processing
Received 0 broadcast packets, 0 multicast packets
Input errors: 0 CRC, 0 overrun, 0 alignment, 0 length, 0 collision
440272 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
Output errors: 0 underruns, 0 aborts, 0 loss of carrier

```

## Troubleshooting the Catalyst Switch

To troubleshoot the Catalyst switch running Cisco IOS software, perform the following procedure.



### Note

---

This procedure is supported on Cisco CRS-1 Multishelf Systems only.

---

## Restrictions

The external Catalyst switches must operate with the same software version: Cisco IOS software release 12.2 with Sup720 engine. Please note that older hardware can also use the CatOS software, but Cisco strongly recommends using the Cisco IOS software.

## SUMMARY STEPS

1. Check that all external GE switches are connected to the RPs and SCs.
2. Connect to the external Catalyst switch.
3. **enable**
4. **show running-config**
5. Verify communication between the Catalyst switch and the RPs and SCs.
6. Contact Cisco Technical Support if the problem is not resolved



## DETAILED STEPS

	Command or Action	Purpose
Step 1	Check that all external GE switches are connected to the RPs and SCs as illustrated in <a href="#">Figure 6-2</a> .	Ensures that the connections for the control plane Ethernet network are made. See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on system control plane Ethernet network to RP and SC cabling.
Step 2	Connect to the console port of an external Catalyst switch.	Provides connection to the control plane Ethernet network. See <i>Cisco CRS-1 Carrier Routing System Multishelf System Interconnection and Cabling Guide</i> for details on connecting to the external switch.
Step 3	<b>enable</b>  <b>Example:</b> CAT6k-1# enable	Enters enter privileged EXEC mode.
Step 4	<b>show running-config</b>  <b>Example:</b> CAT6k-1# show running-config	Displays the contents of the currently running configuration file on the external Catalyst switch.  Check the running configuration to ensure it is the expected configuration. Also, check the following: <ul style="list-style-type: none"> <li>Spanning tree is enabled</li> <li>There is one VLAN in the spanning tree configuration</li> <li>All GE ports are forwarding</li> <li>The external Catalyst switch is a spanning tree root (MAC address indicated for designated root is the same as the bridge ID MAC ADDR)</li> </ul>
Step 5	Verify communication between the Catalyst switch and the RPs and SCs.	See <i>Cisco CRS-1 Carrier Routing System Multishelf System Upgrade and Conversion Guide</i> for information on verifying communication between the Catalyst switch and the RPs and SCs.
Step 6	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the “ <a href="#">Obtaining Documentation and Submitting a Service Request</a> ” section on page viii in the <a href="#">Preface</a> .

## Troubleshooting the CRS-1, 4-slot, 8-slot, or 16-slot System Router Topology

To verify and troubleshoot the Cisco CRS-1 8-Slot Line Card Chassis, Cisco CRS-1 8-Slot Line Card Chassis, or Cisco CRS-1 16-Slot Line Card Chassis router topology, perform the following procedures.

Each SC and RP has two internal switches (0 and 1). Each internal switch has two Gigabit Ethernet control ports. Port 1 is on the front panel and is used as a control port for interchassis connectivity (Gigabit Ethernet control). Port 2 is used to connect between the two internal switches in the SC. The Fast Ethernet (FE) ports of the switches are used within the chassis to connect to other nodes (intrachassis connections).



**Note** Only the Cisco CRS-1 16-Slot Line Card Chassis contains switch 0 and switch 1. The Cisco CRS-1 8-Slot Line Card Chassis and Cisco CRS-1 4-Slot Line Card Chassis contain switch 0.

### SUMMARY STEPS

1. **show platform**
2. **show controllers switch 0 ports location *node-id***
3. **show controllers switch 1 ports location *node-id***
4. **show controllers switch 0 statistics location *node-id***
5. **show controllers switch 1 statistics location *node-id***
6. **show controllers backplane ethernet detail location *node-id***
7. Place the DSC in ROMMON mode:
  - a. **admin**
  - b. **config register 0x0**
  - c. **exit**
  - d. **reload**
8. Exit ROMMON mode:
  - a. **confreg 0x102**
  - b. **reset**
9. Contact Cisco Technical Support if the problem is not resolved

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show platform</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show platform	Displays information and status on each node in the system.  Verify that all components in the router are visible and that the components are in the run state.  If components are not visible in the command output, proceed to <a href="#">Step 3</a> and check the GE switch connections as illustrated in <a href="#">Figure 6-2</a> .
Step 2	<b>show controllers switch 0 ports location <i>node-id</i></b>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 0 ports location 0/RP1/CPU0	Displays status on a Switch Port 0.  Verify that all ports are in the forwarding state.

	Command or Action	Purpose
Step 3	<b>show controllers switch 1 ports location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 1 ports location 0/RP1/CPU0	Displays status on a switch port 1.  <b>Note</b> Only the Cisco CRS-1 16-Slot Line Card Chassis contains switch 0 and switch 1. The Cisco CRS-1 8-Slot Line Card Chassis and Cisco CRS-1 4-Slot Line Card Chassis contain switch 0.  Verify that all ports are in the forwarding state.
Step 4	<b>show controllers switch 0 statistics location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 0 statistics location 0/RP1/CPU0	Displays statistics on Switch Port 0.  Verify that the expected internal switch ports are carrying traffic (Tx and Rx frames).
Step 5	<b>show controllers switch 1 statistics location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers switch 1 statistics location 0/RP1/CPU0	Displays statistics on Switch Port 1.  <b>Note</b> Only the Cisco CRS-1 16-Slot Line Card Chassis contains switch 0 and switch 1. The Cisco CRS-1 8-Slot Line Card Chassis and Cisco CRS-1 4-Slot Line Card Chassis contain switch 0.  Verify that the expected internal switch ports are carrying traffic (Tx and Rx frames).
Step 6	<b>show controllers backplane ethernet detail</b> location <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show controllers backplane ethernet detail location 0/RP1/CPU0	Displays detailed information about backplane interfaces in a specific location.
Step 7	<b>admin</b> <b>config-register 0x0</b> <b>exit</b> <b>reload</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin RP/0/RP0/CPU0:router(admin)# config-register 0x0 RP/0/RP0/CPU0:router(admin)# exit RP/0/RP0/CPU0:router# reload	Places the DSC in ROMMON mode.

	Command or Action	Purpose
Step 8	<b>confreg 0x102</b> <b>reset</b>  <b>Example:</b> rommon B2 > confreg 0x102 rommon B3 > reset	Exits ROMMON mode, resetting and initializing the router.
Step 9	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The output from the **show platform** command indicates that all expected nodes are in the IOS XR RUN state:

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

Node	Type	PLIM	State	Config State
0/1/CPU0	MSC	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/1/0	MSC (SPA)	4XOC3-POS	OK	PWR, NSHUT, MON
0/1/4	MSC (SPA)	4XOC48-POS	OK	PWR, NSHUT, MON
0/1/5	MSC (SPA)	8X1GE	OK	PWR, NSHUT, MON
0/3/CPU0	MSC	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/3/4	MSC (SPA)	8X1GE	OK	PWR, NSHUT, MON
0/6/CPU0	MSC	Jacket Card	IOS XR RUN	PWR, NSHUT, MON
0/6/0	MSC (SPA)	4XOC3-POS	OK	PWR, NSHUT, MON
0/6/4	MSC (SPA)	8XOC3/OC12-POS	OK	PWR, NSHUT, MON
0/6/5	MSC (SPA)	8X1GE	OK	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

The following example shows that the appropriate ports on Switch Port 0 are forwarding:

```
RP/0/RP0/CPU0:router# show controllers switch 0 ports location 0/RP1/CPU0
```

```
Ports Active on Switch 0
FE Port 0 : Up, STP State : FORWARDING (Connects to - 0/RP0)
FE Port 1 : Up, STP State : FORWARDING (Connects to - 0/RP1)
FE Port 2 : Up, STP State : FORWARDING (Connects to - 0/SM0)
FE Port 3 : Up, STP State : FORWARDING (Connects to - 0/SM1)
FE Port 4 : Up, STP State : FORWARDING (Connects to - 0/SM2)
FE Port 5 : Up, STP State : FORWARDING (Connects to - 0/SM3)
FE Port 6 : Down (Connects to - )
FE Port 7 : Down (Connects to - )
FE Port 8 : Down (Connects to - 0/LC0)
FE Port 9 : Up, STP State : FORWARDING (Connects to - 0/LC1)
FE Port 10 : Down (Connects to - 0/LC2)
FE Port 11 : Up, STP State : FORWARDING (Connects to - 0/LC3)
FE Port 12 : Down (Connects to - 0/LC4)
FE Port 13 : Down (Connects to - 0/LC5)
FE Port 14 : Up, STP State : FORWARDING (Connects to - 0/LC6)
FE Port 15 : Down (Connects to - 0/LC7)
GE Port 0 : Down (Connects to - GE_0)
GE Port 1 : Down (Connects to - GE_1)
```

The following example shows the statistics for Switch Port 0:

```
RP/0/RP0/CPU0:router# show controllers switch 0 statistics location 0/RP1/CPU0
```

Port	Tx Frames	Tx Errors	Rx Frames	Rx Errors	Connects
0 :	670451	1	236107	46	0/RP0
1 :	420259	1	876822	64	0/RP1
2 :	865909	0	922949	66	0/SM0
3 :	864762	2049	789500	591	0/SM1
4 :	865011	1877	789172	325	0/SM2
5 :	870186	1	796699	66	0/SM3
6 :	0	512	0	3133	
7 :	0	513	1	5	
8 :	33	0	0	1103	0/LC0
9 :	234601	1	254274	1671	0/LC1
10 :	1025	0	1	421	0/LC2
11 :	84842	257	104034	461	0/LC3
12 :	0	0	0	1063	0/LC4
13 :	0	1	0	0	0/LC5
14 :	146808	0	172052	599	0/LC6
15 :	0	0	0	16	0/LC7
24 :	29250	0	1499184	632	GE_0
25 :	0	0	32784	32784	GE_1

The following example shows output from the **show controller backplane ethernet location detail** command:

```
RP/0/RP0/CPU0:router# show controllers backplane ethernet detail location 0/1/0
```

```
FastEthernet0_RP0_CPU0 is up
  Hardware is 10/100 Ethernet, H/W address is 5246.4800.0201
  Internet address is 10.0.2.1
  MTU 1514 bytes
  Encapsulation HFRIES (HFR Internal Ethernet Server)
  Mode : Full Duplex, Rate : 100Mb/s
    68732206 packets input, 3398506674 bytes, 0 total input drops
    0 packets discarded (0 bytes) in garbage collection
    267 packets discarded (95502 bytes) in rcv processing
    0 incomplete frames discarded
    0 packets discarded due to bad headers
    2 packets waiting for clients
    0 packets waiting on Rx
  Received 16 broadcast packets, 14516874 multicast packets
  Input errors: 0 CRC, 0 overrun, 0 alignment, 0 length, 0 collision
  60733356 packets output, 1676989930 bytes, 0 total output drops
  Output 6904806 broadcast packets, 6904806 multicast packets
  Output errors: 0 underruns, 0 aborts, 0 loss of carrier
  Write rejects : 0
```





# CHAPTER 7

## Collecting System Information

This chapter describes techniques that you can use to collect system information for troubleshooting routers using Cisco IOS XR software. It includes the following sections:

- [Capturing Logs, page 7-165](#)
- [Using ping and traceroute, page 7-166](#)
- [Using Debug Commands, page 7-166](#)
- [Using Diagnostic Commands, page 7-166](#)
- [Commands Used to Display Process and Thread Details, page 7-169](#)

### Capturing Logs

See the “[Troubleshooting Techniques and Approaches](#)” section on page 1-1 in Chapter 1, “[General Troubleshooting](#),” for information on collecting current system information. You can collect system information using the following commands:

- The following commands are used to capture logs:
  - **show tech-support**—Displays system information for Cisco Technical Support
  - **show logging**—Displays the contents of the logging buffers
  - **show system verify**—Displays system verification information
- **dumpplaneeprom**: displays the serial number of the chassis. This command is executed in ROM monitor (ROMMON) mode. The following example shows the output of the command:

```
rommon B2 > dumpplaneeprom

EEPORM data backplane
000000 ff 00 01 e0 ff ff ff ff ff ff ff ff ff ff .....
000010 ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000020 ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000030 ff ff ff ff ff ff 08 00 45 3a 2d 01 04 00 ff ff .....E:-.....
000040 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000050 54 42 43 30 37 31 39 30 31 37 33 30 30 30 30 TBC0719017300000
000060 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000070 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000080 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
000090 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0000a0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0000b0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0000c0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
```

```

0000d0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0000e0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....
0000f0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff .....

```

The "TBC0719017300000" string is the rack number. This string should be present for every chassis. The number is burnt in by manufacturing.

## Using ping and traceroute

For details on the **ping** and **traceroute** commands, see the “Basic Troubleshooting Commands” section in *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router*.

## Using Debug Commands

For details on using **debug** commands, see *Cisco IOS XR Using Debug Guide*.

## Using Diagnostic Commands

The Cisco IOS XR diagnostic tests verify control Ethernet and fabric data paths between nodes in a system using Cisco IOS XR software. If a diagnostic tests fails, it indicates a bad data path. The integrity of the covered data paths is verified when the diagnostic tests pass.

The system runs diagnostic tests automatically to verify control Ethernet and fabric data paths between nodes in the system. The integrity of the covered data paths is verified when the diagnostic tests pass. If a diagnostic tests fails, it indicates a bad data path and the system alerts you to the problem.

The diagnostic tests generally test data paths between multiple nodes, therefore you need to analyze error reports to narrow down the possible points of failure in a system. For example, if a diagnostic test fails on a fabric interface, the fabric cables in the path of failure would be a primary suspected cause of the failure.

All diagnostic tests run within the 1 second to 1 minute range.

This section contains the following additional topics related to diagnostics:

- [Online Diagnostics, page 7-166](#)
- [Transient Condition when Standby RP Becomes Active, page 7-168](#)
- [Offline Diagnostics—FDIAG RUNNING State, page 7-169](#)
- [Additional Reference for Diagnostic Commands, page 7-169](#)

## Online Diagnostics

You can start and stop diagnostic tests on specific nodes while the node is online and processing traffic. It is important to run test on both the active and standby RP; the standby RP is actually capable of running more fabric diagnostic tests than the active RP.

### Examples:

The following example shows a set of diagnostic tests on the active RP (0/RP0/CPU0).



```
RP/0/RP0/CPU0:router(admin)#diagnostic start location 0/RP0/CPU0 test non-disruptive

Wed Sep 1 12:50:24.156 PDT
RP/0/RP0/CPU0:Sep 1 12:50:24.426 : online_diag_rp[351]:
%DIAG-DIAG-6-TEST_SKIPPED_FROM_ACTIVE : RP 0/RP0/CPU0: ControlEthernetInactiveLinkTest
cannot be executed from active node.
RP/0/RP0/CPU0:Sep 1 12:50:24.426 : online_diag_rp[351]:
%DIAG-DIAG-6-TEST_SKIPPED_FROM_ACTIVE : RP 0/RP0/CPU0: FabricDiagnosisTest cannot be
executed from active node.
RP/0/RP0/CPU0:Sep 1 12:50:24.428 : online_diag_rp[351]:
%DIAG-DIAG-6-TEST_SKIPPED_FROM_ACTIVE : RP 0/RP0/CPU0: FabricMcastTest cannot be
executed from active node.
RP/0/RP0/CPU0:Sep 1 12:50:24.430 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running ControlEthernetPingTest{ID=1} ...
RP/0/RP0/CPU0:Sep 1 12:50:28.703 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: ControlEthernetPingTest{ID=1} has completed successfully
```

```
RP/0/RP0/CPU0:router(admin)#show diagnostic result location 0/RP0/CPU0
```

```
Wed Sep 1 12:51:41.606 PDT

Current bootup diagnostic level for RP 0/RP0/CPU0: bypass
RP 0/RP0/CPU0:
Overall diagnostic result: MINOR ERROR
Diagnostic level at card bootup: bypass

Test results: (. = Pass, F = Fail, U = Untested)

1 ) ControlEthernetPingTest -----> .
2 ) SelfPingOverFabric -----> .
3 ) FabricPingTest -----> .
4 ) ControlEthernetInactiveLinkTest -> U
5 ) RommonRevision -----> F
6 ) FabricDiagnosisTest -----> U
7 ) FilesystemBasicDisk0 -----> .
8 ) FilesystemBasicDisk1 -----> .
9 ) FilesystemBasicHarddisk -----> .
10 ) ScratchRegisterTest:

Device 1 2 3 4
-----
      . . . .

11 ) FabricMcastTest -----> U
```

The following example shows a set of diagnostic tests on the standby RP (0/RP1/CPU0).

```
RP/0/RP0/CPU0:router(admin)#diagnostic start location 0/RP1/CPU0 test non-disruptive

Wed Sep 1 12:50:52.703 PDT
RP/0/RP1/CPU0:Sep 1 12:50:54.242 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP1/CPU0: Running ControlEthernetPingTest{ID=1} ...
RP/0/RP1/CPU0:Sep 1 12:50:58.686 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP1/CPU0: ControlEthernetPingTest{ID=1} has completed successfully
RP/0/RP1/CPU0:Sep 1 12:50:58.686 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP1/CPU0: Running SelfPingOverFabric{ID=2} ...
RP/0/RP0/CPU0:Sep 1 12:50:58.809 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: SelfPingOverFabric{ID=2} has completed successfully
RP/0/RP0/CPU0:Sep 1 12:50:58.809 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running FabricPingTest{ID=3} ...
RP/0/RP0/CPU0:Sep 1 12:51:00.672 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: FabricPingTest{ID=3} has completed successfully
```

```

RP/0/RP0/CPU0:Sep  1 12:51:00.672 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running FilesystemBasicDisk0{ID=7} ...
RP/0/RP0/CPU0:Sep  1 12:51:00.697 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: FilesystemBasicDisk0{ID=7} has completed successfully
RP/0/RP0/CPU0:Sep  1 12:51:00.697 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running FilesystemBasicDisk1{ID=8} ...
RP/0/RP0/CPU0:Sep  1 12:51:00.749 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: FilesystemBasicDisk1{ID=8} has completed successfully
RP/0/RP0/CPU0:Sep  1 12:51:00.749 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running FilesystemBasicHarddisk{ID=9} ...
RP/0/RP0/CPU0:Sep  1 12:51:01.796 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: FilesystemBasicHarddisk{ID=9} has completed successfully
RP/0/RP0/CPU0:Sep  1 12:51:01.796 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running ScratchRegisterTest{ID=10} ...
RP/0/RP0/CPU0:Sep  1 12:51:02.799 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_OK : RP
0/RP0/CPU0: ScratchRegisterTest{ID=10} has completed successfully
RP/0/RP0/CPU0:Sep  1 12:51:02.799 : online_diag_rp[351]: %DIAG-DIAG-6-TEST_RUNNING :
RP 0/RP0/CPU0: Running RommonRevision{ID=5} ...
RP/0/RP0/CPU0:Sep  1 12:51:02.800 : online_diag_rp[351]: %DIAG-DIAG-3-TEST_FAIL : RP
0/RP0/CPU0: RommonRevision{ID=5} has failed. Error code = 0x1 (DIAG_FAILURE)

```

```
RP/0/RP0/CPU0:router(admin)#show diagnostic result location 0/RP1/CPU0
```

```
Wed Sep  1 13:03:28.617 PDT
```

```
Current bootup diagnostic level for RP 0/RP1/CPU0: bypass
```

```
RP 0/RP1/CPU0:
```

```
Overall diagnostic result: MINOR ERROR
```

```
Diagnostic level at card bootup: bypass
```

```
Test results: (. = Pass, F = Fail, U = Untested)
```

```

1 ) ControlEthernetPingTest -----> .
2 ) SelfPingOverFabric -----> .
3 ) FabricPingTest -----> .
4 ) ControlEthernetInactiveLinkTest -> .
5 ) RommonRevision -----> F
6 ) FabricDiagnosisTest -----> .
7 ) FilesystemBasicDisk0 -----> .
8 ) FilesystemBasicDisk1 -----> .
9 ) FilesystemBasicHarddisk -----> .
10 ) ScratchRegisterTest:

```

```
Device 1 2 3 4
```

```
-----
```

```
. . . .
```

```
11 ) FabricMcastTest -----> .
```

## Transient Condition when Standby RP Becomes Active

If online diagnostics are performed within five minutes of the standby RP becoming active, some test cases will be skipped. Wait at least five minutes after the standby RP is ready before performing the online diagnostic test. If your system is set to perform diagnostic checks automatically, it might skip some tests during this five-minute period. Therefore, you should perform these tests manually after the standby RP has been active for at least five minutes.

To run a specified on-demand diagnostic test or series of tests, use the **diagnostic start location** command.

**Examples:**

```
RP/0/RP0/CPU0:router(admin)# diagnostic start location 0/RP1/CPU0 test 1  
RP/0/RP0/CPU0:router(admin)# diagnostic stop location 0/RP1/CPU0
```

## Offline Diagnostics—FDIAG RUNNING State

You load the offline diagnostics with the command (admin)**diagnostic load location node-id**. The specified node remains in the "FDIAG RUNNING" state until you unload diagnostics with the command (admin)**diagnostic unload location node-id**.

**Note**

---

In the "FDIAG RUNNING" state, the specified node is offline and cannot process traffic.

---

While a node is in "FDIAG RUNNING" state, tests are run in response to the optional **autostart** keyword of the **diagnostic load location node-id** command or the **diagnostic start location node-id** command. When an individual test completes, a message is printed and results are updated. The result can be read using the **show diagnostic result location node-id** command. When a test completes, a new **diagnostic start location node-id** command can be invoked since the card remains the "FDIAG RUNNING" state until it is explicitly unloaded using the **diagnostic unload location node-id** command.

## Additional Reference for Diagnostic Commands

For details on diagnostics commands and available tests, see *Cisco IOS XR Diagnostics* at the Configuration Guide site:

[http://www.cisco.com/en/US/products/ps5763/products_installation_and_configuration_guides_list.html](http://www.cisco.com/en/US/products/ps5763/products_installation_and_configuration_guides_list.html)

## Commands Used to Display Process and Thread Details

For details on processes and threads, see the "Understanding Processes and Threads" section in *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router*.





## CHAPTER 8

# Process Monitoring and Troubleshooting

---

This chapter includes the following sections:

- [System Manager, page 8-172](#)
- [Watchdog System Monitor, page 8-172](#)
- [Core Dumps, page 8-173](#)
- [follow Command, page 8-173](#)
- [show processes Commands, page 8-175](#)
- [Redundancy and Process Restartability, page 8-179](#)
- [Process States, page 8-180](#)
- [Process Monitoring, page 8-182](#)
- [Monitoring CPU Usage and Using Syslog Messages, page 8-183](#)
- [Troubleshooting High CPU Utilization and Process Timeouts, page 8-185](#)
- [Troubleshooting a Process Restart, page 8-195](#)

The Cisco IOS XR software is built on a modular system of processes. A process is a group of threads that share virtual address (memory) space. Each process provides specific functionality for the system and runs in a protected memory space to ensure that 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.

Threads are units of execution, each with an execution context that includes a stack and registers. A thread is in effect a “sub-process” managed by the parent, responsible for executing a subportion of the overall process. For example, Open Shortest Path First (OSPF) has a thread which handles “hello” receipt and transmission. A thread may only run when the parent process is allocated runtime by the system scheduler. A process with threads is a multi-threaded process.

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. For more information on processes, see *Cisco IOS XR System Management Configuration Guide for the Cisco CRS-1 Router*.

# System Manager

Each process is assigned a job ID (JID) when started. The JID does not change when a process is started, stopped, then restarted. Each process is also assigned a process ID (PID) when started, but this PID changes each time the process is stopped and restarted.

The System Manager (sysmgr) is the fundamental process and the foundation of the system. The sysmgr is responsible for monitoring, starting, stopping, and restarting almost all processes on the system. The restarting of processes is predefined (respawn flag on or off) and honored by sysmgr. The sysmgr is the parent of all processes started on boot-up and by configuration. Two instances are running on each node providing a hot standby process level redundancy. Each active process is registered with the SysDB and once started by the sysmgr active process the sysmgr is notified when it is running. If the sysmgr active process is dying the standby process takes over the active state and a new standby process is generated.

The sysmgr running on the line card (LC) handles all the system management duties like process creation, re-spawning, and core-dumping relevant to that node.

The sysmgr itself is started on bootup by the initialization process. Once the sysmgr is started, initialization hands over the ownership of all processes started by initialization to sysmgr and exits.

## Watchdog System Monitor

The Watchdog System Monitor (wdsysmon) keeps historical data on processes and posts this information to a fault detector dynamic link library (DLL), which can then be queried by manageability applications. Once per minute, wdsysmon polls the kernel for process data. This data is stored in a database maintained by the fm_fd_wdsysmon.dll fault detector, which is loaded by wdsysmon.

For more information on wdsysmon and memory thresholds, see the [“Watchdog System Monitor” section on page 9-197 in Chapter 9, “Troubleshooting Memory.”](#)

## Deadlock detections

Wdsysmon can attempt to find deadlocks because thread state is returned with the process data. Wdsysmon specifically looks for mutex deadlocks and local Inter-Process Communication (IPC) hangs. Only local IPC deadlocks can be detected. If deadlocks are detected, debugging information is collected in disk0:/wdsysmon_debug.

Deadlocked processes can be stopped and restarted manually using the **processes restart** command.

## Hang detection

When an event manager is created in the system, the event manager library registers the event with wdsysmon. Wdsysmon expects to periodically hear a “pulse” from every registered event manager in the system. When an event manager is missing, wdsysmon runs a debug script that shows exactly what the thread that created the event manager is doing.

# Core Dumps

When a process is abnormally terminated, a core dump file is written to a designated destination. A core dump contains the following information:

- register information
- thread status information
- process status information
- selected memory segments.

Use the **show exception** command to display the configured core dump settings. The output from the show exception command displays the core dump settings configured with the following commands:

- **exception filepath**
- **exception dump-tftp-route**
- **exception kernel memory**
- **exception pakmem**
- **exception sparse**
- **exception sprsize**

The following example shows the core dump settings.

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

```
Choice 1 path = harddisk:/coredump compress = on filename = <process_name.time>
Choice 2 path = tftp://223.255.254.254/users/xyz compress = on filename =
<process_name.time>
Exception path for choice 3 is not configured or removed
Choice fallback one path = harddisk:/dumper compress = on filename = <process_name>
Choice fallback two path = disk1:/dumper compress = on filename = <process_name>
Choice fallback three path = disk0:/dumper compress = on filename = <process_name>
Kernel dump not configured
Tftp route for kernel core dump not configured
Dumper packet memory in core dump enabled
Sparse core dump enabled
Dumper will switch to sparse core dump automatically at size 300MB
```

Coredumps can be generated manually using the **dumpcore** command. There are two types of core dumps that can be manually run:

- running—does not impact services
- suspended—suspends a process while generating the core dump

The **show context** command shows the coredump information for the last 10 core dumps

## follow Command

The **follow** command is used to unobtrusively debug a live process or live thread in a process. The follow command is particularly useful for:

- process deadlock, livelock, or mutex conditions
- high CPU use conditions

- examining the contents of a memory location or a variable in a process to determine the cause of a corruption issue
- investigating issues where a process or thread is stuck in a loop.

A livelock condition is where two or more processes continually change their state in response to changes in the other processes.

The following actions can be specified with the **follow** command:

- Follow all live threads of a given process or a given thread of a process and print stack trace in a format similar to core dump output
- Follow a process in a loop for a given number of iterations
- Set a delay between two iterations while invoking the command
- Set the priority at which this process should run while this command is being executed
- Dump memory from a given virtual memory location for a given size
- Display register values and status information of the target process
- Take a snapshot of the execution path of a thread asynchronously to investigate performance-related issues – this can be done by specifying a high number of iterations with a zero delay



#### Caution

If your system is running Release 3.8.0, 3.8.1, 3.8.2, or 3.9.0 software, you should *not* run the **follow process** and **follow job** commands, because these can cause a kernel crash at the target node. Therefore, for these software releases, you should use other available commands for troubleshooting and call Cisco Technical Support if the problem is not resolved. (This crash behavior does not occur for releases other than the ones listed.)

The following example shows the live thread of process 929034375.

```
RP/0/RP0/CPU0:router# follow process 929034375
```

```
Attaching to process pid = 929034375 (pkg/bin/bgp)
No tid specified, following all threads
```

```
DLL Loaded by this process
```

```
-----
```

DLL path	Text addr.	Text size	Data addr.	Data size	Version
/pkg/lib/libsysmgr.dll	0xfc122000	0x0000df0c	0xfc0c2b14	0x000004ac	0
/pkg/lib/libcerrno.dll	0xfc130000	0x00002f04	0xfc133000	0x00000128	0
/pkg/lib/libcerr_dll_tbl.dll	0xfc134000	0x00004914	0xfc133128	0x00000148	0
/pkg/lib/libltrace.dll	0xfc139000	0x00007adc	0xfc133270	0x00000148	0
/pkg/lib/libinfra.dll	0xfc141000	0x00033c90	0xfc1333b8	0x00000bbc	0
/pkg/lib/libcerrno/libinfra_error.dll	0xfc1121dc	0x00000cd8	0xfc175000	0x000000a8	0
/pkg/lib/libbios.dll	0xfc176000	0x0002dab0	0xfc1a4000	0x00002000	0
/pkg/lib/libcerrno/libevent_manager_error.dll	0xfc1a6000	0x00000e88	0xfc133f74	0x00000000	0
/pkg/lib/libc.dll	0xfc1a7000	0x00079d70	0xfc221000	0x00002000	0
/pkg/lib/libsyslog.dll	0xfc223000	0x000054e0	0xfc1750a8	0x00000328	0
/pkg/lib/libplatform.dll	0xfc229000	0x0000c25c	0xfc236000	0x00002000	0
/pkg/lib/libbackplane.dll	0xfc243000	0x000013a8	0xfc1755b8	0x000000a8	0
/pkg/lib/libcerrno/libpkgfs_error.dll	0xfc245000	0x00000efc	0xfc175660	0x00000088	0
/pkg/lib/libnodeid.dll	0xfc246000	0x0000a588	0xfc1756e8	0x00000248	0
/pkg/lib/libdebug.dll	0xfc29b000	0x0000fdb0	0xfc2ab000	0x00000570	0
/pkg/lib/libcerrno/libdebug_error.dll	0xfc294244	0x00000db0	0xfc175c68	0x000000e8	0
/pkg/lib/lib_procfs_util.dll	0xfc2ac000	0x00004f20	0xfc175d50	0x000002a8	0
/pkg/lib/libinst_debug.dll	0xfc375000	0x0000357c	0xfc36d608	0x000006fc	0
/pkg/lib/libpackage.dll	0xfc3c8000	0x00041ad0	0xfc40a000	0x00000db4	0



```

/pkg/lib/libwd_evm.dll    0xfc40b000 0x00003dc4 0xfc36dd04 0x00000168      0
.
.
.
Iteration 1 of 5
-----

Current process = "pkg/bin/bgp", PID = 929034375 TID = 1

trace_back: #0 0xfc164210 [MsgReceivev]
trace_back: #1 0xfc14ecb8 [msg_receivev]
trace_back: #2 0xfc14eac4 [msg_receive]
trace_back: #3 0xfc151f98 [event_dispatch]
trace_back: #4 0xfc152154 [event_block]
trace_back: #5 0xfd8e16a0 [bgp_event_loop]
trace_back: #6 0x48230db8 [<N/A>]
trace_back: #7 0x48201080 [<N/A>]

ENDOFSTACKTRACE

Current process = "pkg/bin/bgp", PID = 929034375 TID = 2

trace_back: #0 0xfc164210 [MsgReceivev]
trace_back: #1 0xfc14ecb8 [msg_receivev]
trace_back: #2 0xfc14eac4 [msg_receive]
trace_back: #3 0xfc151f98 [event_dispatch]
trace_back: #4 0xfc152154 [event_block]
trace_back: #5 0xfc50efd8 [chk_evm_thread]

ENDOFSTACKTRACE
.
.
.

```

## show processes Commands

The following show processes commands are used to display process information:

- [show processes boot Command, page 8-175](#)
- [show processes startup Command, page 8-176](#)
- [show processes failover Command, page 8-177](#)
- [show processes blocked Command, page 8-178](#)

## show processes boot Command

The **show processes boot** command displays process boot information. Use the command output to check the following:

- How long it took the processes to start
- The order that the processes started
- Was a process delayed indicating a boot failure or boot problems
- Did the processes start within the time constraints set by the system

```
RP/0/RP0/CPU0:router# show processes boot location 0/rp1/cpu0
```

```

Band Name           Finished    %Idle      JID    Ready Last Process
-----
 1.0 MBI             22.830    65.130%    62    22.830 insthelper
40.0 ARB             129.225    92.080%    154   106.395 dsc
90.0 ADMIN           185.140     5.950%    175   55.915 fabricq_mgr
100.0 INFRA          207.372    25.040%    165   22.232 fib_mgr
150.0 STANDBY        231.605    13.840%    104   24.233 arp
999.0 FINAL          237.942     1.590%    234     6.337 ipv6_rump

Started Level      JID Inst    Ready Process
-----
 0.000s  0.05        80    1    0.000 wd-mbi
 0.000s  1.00        57    1    0.000 dllmgr
 0.000s  2.00        71    1    0.000 pkgfs
 0.000s  3.00        56    1    0.000 devc-conaux
 0.000s  3.00        73    1    0.000 devc-pty
 0.000s  6.00        70    1    0.000 pipe
.
.
.
Last process started:    6d19h after boot. Total: 174

```

## show processes startup Command

The **show processes startup** command displays process data for processes created at startup. Use the command output to check the following:

- Are the listed processes, including their state, start time, restart status, placement, and mandatory status as expected
- How long it took the processes to start
- The order in which the processes started
- Was a process delayed indicating a boot failure or boot problems
- Did the processes start within the time constraints set by the system

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

```

JID    LAST STARTED          STATE    RE-    PLACE-  MANDA-  NAME(IID)  ARGS
-----
81     07/05/2006 14:46:37.514 Run      1              M        wd-mbi(1)
57     07/05/2006 14:46:37.514 Run      1              M        dllmgr(1) -r 60
-u 30
72     07/05/2006 14:46:37.514 Run      1              M        pkgfs(1)
56     07/05/2006 14:46:37.514 Run      1              M        devc-conaux(1) -
h -d librs232.dll -m libconaux.dll -u libst16550.dll
74     07/05/2006 14:46:37.514 Run      1              M        devc-pty(1) -n 3
2
55     Not configured        None     0              M        clock_chip(1) -r
-b
71     07/05/2006 14:46:37.514 Run      1              M        pipe(1)
65     07/05/2006 14:46:37.514 Run      1              M        mqueue(1)
64     Not configured        None     0              M        cat(1) /etc/motd
73     Not configured        None     0              M        platform_dllmap(
1)
77     07/05/2006 14:46:37.514 Run      1              M        shmwin_svr(1)
60     07/05/2006 14:46:37.514 Run      1              M        devf-scrp(1) -e
0xf0000038 -m /bootflash: -s 0xfc000000,64m -r -t4 -b10

```

```

66      Not configured      None      0          M      nname(1)
69      07/05/2006 14:46:37.514 Run      1          M      pci_bus_mgr(1) -
o
288     07/05/2006 14:47:02.799 Run      1          M      qsm(1)
68      07/05/2006 14:46:37.514 Run      1          M      obflmgr(1)
.
.
.
-----
Total pcbs: 198

```

## show processes failover Command

The **show processes failover** command displays process failover information. The command output displays information on how long it took processes to start after a failover (node reboot). Check if there were any delays.

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

```

Thu May  3 11:16:05.562 EST EDT
Band Name          Finished      %Idle      JID    Ready Last Process
-----
40.0 ARB            0.000      0.000%      0      0.000 NONE
90.0 ADMIN          0.000      0.000%      0      0.000 NONE
100.0 INFRA         0.000      0.000%      0      0.000 NONE
121.0 FT_ADMIN      0.056      0.000%     315     0.056 qsm
122.0 FT_INFRA      1.819      0.000%     195     1.763 ifmgr
123.0 FT_IP_ARM     2.097      0.000%     232     0.278 ipv6_arm
124.0 FT_ISIS       2.280      0.000%     248     0.183 isis
125.0 FT_PRE_IP     2.303      0.000%      0      0.023 NONE
126.0 FT_IP         6.345      0.000%     219     4.042 ipv4_local
127.0 FT_LPTS       8.041      0.000%     262     1.696 lpts_pa
128.0 FT_PRE OSPF    9.889      0.000%     260     1.848
loopback_caps_partner
129.0 FT OSPF       17.944     37.410%     291     8.055 ospf
130.0 FT MPLS       23.602      0.000%     272     5.658 mpls_lsd
131.0 FT_BGP_START  26.366      0.000%     326     2.764 rsvp
132.0 FT_MULTICAST  32.940      0.000%     277     6.574 mrib
133.0 FT_CLI        35.357      0.000%     361     2.417
tty_session_startup
134.0 FT_FINAL      44.772      0.000%     322     9.415
rip_policy_reg_agent
150.0 ACTIVE        70.322      0.000%     224    25.550 ntpd
999.0 FINAL         79.011      0.000%     273     8.689 mpls_rid_helper

```

```

Go active  Level Band Name          JID Inst  Avail Process
-----
0.002s    22.00 FT_ADMIN          315    1    0.049 qsm
0.027s    38.00 FT_ADMIN           52    1    0.000 bcm_process
0.028s    40.00 FT_ADMIN          155    1    0.012 dsc
0.028s    85.00 FT_ADMIN          332    1    0.000 shelfmgr
0.030s    85.00 FT_ADMIN          333    1    0.000 shelfmgr_partner
0.031s   120.00 FT_ADMIN          184    1    0.000 fab_svr
0.061s    23.00 FT_INFRA          145    1    0.000 correlatord
0.064s    23.00 FT_INFRA          352    1    0.000 syslogd
0.065s    23.00 FT_INFRA           79    1    0.000 syslogd_helper
0.066s    38.00 FT_INFRA          297    1    0.000 packet
0.067s    40.00 FT_INFRA          379    2    0.000 chkpt_proxy
0.070s    40.00 FT_INFRA          380    3    0.000 chkpt_proxy
0.072s    40.00 FT_INFRA          381    4    0.000 chkpt_proxy
.

```

```

.
.
85.006s  0.00      368    1    2.094 udp_snmpd
      85.310s  0.00      373    1   25.730 vrrp
      85.743s  0.00      376    1   22.666 xmlagent
      7d07h   0.00      136    1    0.479 chdlc_ma

Last process started:    7d07h after switch over. Total: 78

```

## show processes blocked Command

The **show processes blocked** command displays details about reply, send, and mutex blocked processes.

Since a temporary blocked state for any process is possible, it is recommended to run the **show processes blocked** command two times consecutively for each interval and for each node. If a process is displayed as blocked after the first and second iteration, you can run the command a third time to ensure the process is blocked.

The polling interval should not be too short (enough to show a sustained blocked state). For example, the Cisco CRS-1 8-Slot Line Card Chassis requires a minimum of 20 requests for each interval (2 RPs and 8 LCs) if fully equipped.

The **show processes blocked** command output always displays processes in the Reply state as blocked.

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

```

Wed May  2 11:44:12.360 EST EDT
  Jid      Pid Tid      Name State Blocked-on
65546      8202  1      ksh Reply    8200 devc-conaux
   52      36889  2    attachd Reply   32791 eth_server
   52      36889  3    attachd Reply   12301 mqueue
   77      36891  6      qnet Reply   32791 eth_server
   77      36891  7      qnet Reply   32791 eth_server
   77      36891  8      qnet Reply   32791 eth_server
   77      36891  9      qnet Reply   32791 eth_server
   51      36897  2    attach_server Reply   12301 mqueue
  376     139341  1    tftp_server Reply   12301 mqueue
  364     143438  6    sysdb_mc Reply  135244 gsp
  268     221354  2      lpts_fm Reply  204855 lpts_pa
65725    13291709  1      exec Reply     1 kernel
65784    23720184  1      exec Reply   331975 devc-vty
65786    27287802  1      exec Reply   331975 devc-vty
65788    23589116  1      attach Reply    8200 devc-conaux
65788    23589116  2      attach Reply   12301 mqueue
65790    27316478  1      exec Reply     1 kernel
65792    27328768  1      exec Reply   331975 devc-vty
65793    27726081  1      more Reply   12299 pipe
   350     27418882  2      snmpd Reply   143438 sysdb_mc
   385     27418886  1    udp_snmpd Reply   221353 udp
65800    27726088  1    show_processes Reply     1 kernel

```

For these processes it is a normal output. For example, the line:

```
65770 27726088    1    show_processes Reply     1 kernel
```

is a direct result of executing the **show processes blocked** command. Each time the command is applied the process ID (PID) will change.

If a vital system process or fundamental application controlling connectivity (for example, routing protocols or Multiprotocol Label Switching Label Distribution Protocol [MPLS LDP]) appears blocked in the Reply, Sent, Mutex, or Condvar state, do the following:

- Collect data from the **follow job** or **follow process** command. See the “[follow Command](#)” section on [page 8-173](#) for more information on these commands.

**Caution**

If your system is running Release 3.8.0, 3.8.1, 3.8.2, or 3.9.0 software, you should *not* run the **follow process** and **follow job** commands, because these can cause a kernel crash at the target node. Therefore, for these software releases, you should use other available commands for troubleshooting and call Cisco Technical Support if the problem is not resolved. (This crash behavior does not occur for releases other than the ones listed.)

- Use the **dumpcore running job-id location node-id** command on the affected process. The output of the dumpcore is located in `harddisk:/dumper`, unless the location has been configured using the **exception choice** command.

**Caution**

Some processes are dangerous to restart. It is recommended that you involve your technical representative and follow the advice from Cisco Technical Support. For contact information for Cisco Technical Support, see the “[Obtaining Documentation and Submitting a Service Request](#)” section on [page viii](#) in the [Preface](#).

**Example:**

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

Jid	Pid	Tid	Name	State	Blocked-on
65546	8202	1	ksh	Reply	8200 devc-conaux
51	36890	2	attachd	Reply	32791 eth_server
51	36890	3	attachd	Reply	12301 mqueue
75	36893	5	qnet	Reply	32791 eth_server
75	36893	6	qnet	Reply	32791 eth_server
75	36893	7	qnet	Reply	32791 eth_server
75	36893	8	qnet	Reply	32791 eth_server
50	36899	2	attach_server	Reply	12301 mqueue
334	172108	1	tftp_server	Reply	12301 mqueue
247	290991	2	lpts_fm	Reply	184404 lpts_pa
65750	644260054	1	exec	Reply	1 kernel
65752	655270104	1	config	Reply	286888 devc-vty
367	2642149	5	mpls_ldp	Reply	2642153 lspv_server
65772	655229164	1	exec	Reply	1 kernel
65773	656842989	1	more	Reply	12299 pipe
65774	656842990	1	show_processes	Reply	1 kernel

**Note**

To troubleshoot a blocked process, use the procedure in the “[Troubleshooting a Process Block](#)” section on [page 8-188](#).

## Redundancy and Process Restartability

On systems using Cisco IOS XR software, applications primarily use a combination of two fatal error recovery methods: process restartability and process (application) redundancy.

Process restart is typically used as the first level of failure recovery. If the checkpointed data is not corrupted, the crashed process can recover after it is restarted. If multiple restarts of a mandatory process fail or if peer processes cannot recover from a crashed process restart the standby card becomes active.

For a non-mandatory process, if the number of respawns per minute is reached then the sysmgr stops to restart the process and the application has to be restarted manually.

Each process not triggered by configuration is, by default, started as 'mandatory' (critical for router to function) process. If a mandatory process crashes five times within a five minute window, an RP switchover is triggered if the standby RP is ready. The **show processes all** command lists all processes and process state including mandatory flag. The mandatory flag can be switched OFF. The **process mandatory {on | off} {executable-name | job-id} [location node-id]** command is used to switch on and off the mandatory flag.

## Process States

Within the Cisco IOS XR software there are servers that provide the services and clients that use the services. A specific process can have a number of threads that provide the same service. Another process can have a number of clients that may require a specific service at any point in time. Access to the servers is not always available, and if a client requests access to a service it will wait for the server to be free. When this happens the client is blocked. The client may be blocked because its waiting for a resource such as a mutex or it may be blocked because the server has not replied.

In the following example, the **show process ospf** command is used to check the status of the threads in the ospf process.

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

      Job Id: 250
      PID: 299228
      Executable path: /disk0/hfr-rout-3.4.0/bin/ospf
      Instance #: 1
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
      Max. spawns per minute: 12
      Last started: Wed Nov  8 15:45:59 2006
      Process state: Run
      Package state: Normal
      Started on config: cfg/gl/ipv4-ospf/proc/100/ord_f/default/ord_a/routerid
      core: TEXT SHARED MEM MAIN MEM
      Max. core: 0
      Placement: ON
      startup_path: /pkg/startup/ospf.startup
      Ready: 3.356s
      Available: 7.363s
      Process cpu time: 2.648 user, 0.186 kernel, 2.834 total

JID   TID   Stack pri state      HR:MM:SS:MSEC NAME
272   1     60K   10 Receive    0:00:00:0563 ospf
272   2     60K   10 Receive    0:00:00:0017 ospf
272   3     60K   10 Receive    0:00:00:0035 ospf
272   4     60K   10 Receive    0:00:02:0029 ospf
272   5     60K   10 Receive    0:00:00:0003 ospf
272   6     60K   10 Condvar    0:00:00:0001 ospf
272   7     60K   10 Receive    0:00:00:0000 ospf
-----
```

The process ospf is given a Job ID of 250. This Job ID never changes on a running router. Within the ospf process there are 7 threads, each with their own Thread ID or TID. For each thread, the stack space for each thread, the priority of each thread, and the thread state is listed. [Table 8-1](#) lists the thread states.

The PID is 299228. This number changes each time the process is restarted. The Respawn count indicates how many times the process has restarted and the Process state should show the RUN state.

## Synchronous Message Passing

The message passing life cycle is as follows:

1. A server creates a message channel.
2. A client connects to a channel of a server (analogous to posix open).
3. A client sends a message to a server (MsgSend) and waits for a reply and blocks.
4. The server receives (MsgReceive) a message from a client, processes the message and replies to the client.
5. The client unblocks and processes the reply from the server.

This blocking client-server model is synchronous message passing. This means the client sends a message and blocks. The server receives the message, processes it, replies back to the client, and then the client unblocks. The specific details are as follows.

1. Server is waiting in RECEIVE state
2. Client sends a message to the server and becomes BLOCKED
3. Server receives the message and unblocks (if waiting in receive state)
4. Client moves to the REPLY state
5. Server moves to the RUNNING state
6. Server processes the message
7. Server replies to the client
8. Client unblocks

Use the **show processes** command to display the states the client and servers are in. [Table 8-1](#) lists the thread states.

## Blocked Processes and Process States

Use the **show processes blocked** command to display the processes that are in blocked state.

Synchronized message passing enables you to track the life cycle of inter-process communication between the different threads. At any point in time a thread can be in a specific state. A blocked state can be a symptom of a problem. This does not mean that if a thread is in blocked state then there is a problem—blocked threads are normal. Using the **show processes blocked** command is sometimes a good way to start troubleshooting operating system-type problems. If there is a problem, for example the CPU is high, then use the **show processes blocked** command to determine if anything looks abnormal (what is not normal for your functioning router). This provides a baseline for you to use as a comparison when troubleshooting process life cycles.

At any point in a time a thread can be in a particular state. [Table 8-1](#) lists the thread states.

**Table 8-1 Thread States**

If the State is:	The Thread is:
DEAD	Dead. The Kernel is waiting to release the threads resources.
RUNNING	Actively running on a CPU.
READY	Not running on a CPU but is ready to run.
STOPPED	Suspended (SIGSTOP signal).
SEND	Waiting for a server to receive a message.
RECEIVE	Waiting for a client to send a message.
REPLY	Waiting for a server to reply to a message.
STACK	Waiting for more stack to be allocate.
WAITPAGE	Waiting for the process manager to resolve a page fault.
SIGSUSPEND	Waiting for a signal.
SIGWAITINFO	Waiting for a signal.
NANOSLEEP	Sleeping for a period of time.
MUTEX	Waiting to acquire a mutex.
CONDVAR	Waiting for a conditional variable to be signaled.
JOIN	Waiting for the completion of another thread.
INTR	Waiting for an interrupt.
SEM	Waiting to acquire a semaphore.

To troubleshoot a blocked process, use the procedure in the [“Troubleshooting a Process Block”](#) section on page 8-188.

## Process Monitoring

Significant events of the sysmgr are stored in /tmp/sysmgr.log. The log is a wrapping buffer and is useful for troubleshooting. Use the **show processes aborts location node-id all** command or the **show sysmgr trace verbose | include PROC_ABORT** command to display an overview of abnormally terminated processes.

Because the sysmgr is already monitoring all processes on the system it is not necessarily required to monitor vital processes by external management tools. But, you can use the **show fault manager metric process pid location node-id** command to check critical processes on a regular basis (for example, twice each day). The command output provides information including the abort behavior and the reason of the particular process.

The following example shows OSPF critical process details. Check the number of times the process ended abnormally and the number of abnormal ends within the past time periods.

```
RP/0/RP0/CPU0:router# show fault manager metric process ospf location

=====
job id: 269, node name: 0/RP0/CPU0
process name: ospf, instance: 1
-----
last event type: process start
```



```

recent start time: Wed Jul  5 15:17:48 2006
recent normal end time: n/a
recent abnormal end time: n/a
number of times started: 1
number of times ended normally: 0
number of times ended abnormally: 2
most recent 10 process start times:
-----
Wed Jul  5 15:17:48 2006
-----

most recent 10 process end times and types:

cumulative process available time: 162 hours 20 minutes 51 seconds 452 milliseconds
cumulative process unavailable time: 0 hours 0 minutes 0 seconds 0 milliseconds
process availability: 1.000000000
number of abnormal ends within the past 60 minutes (since reload): 0
number of abnormal ends within the past 24 hours (since reload): 0
number of abnormal ends within the past 30 days (since reload): 2

```

The vital system processes are: qnet, gsp, qsm, redcon, netio, ifmgr, fgid_aggregator, fgid_server, fgid_allocator, fsdb_server, fsdb_aserver, fabricq_mgr, fia_driver, shelfmgr, and lrd on the RP and fabricq_mgr, ingressq, egressq, pse_driver, fia_driver, cpuctrl, and pla_server on the line card.

It is also important to regularly check if critical or vital processes are in a blocked state. See the [“show processes blocked Command” section on page 8-178](#) for information on checking if processes are in the blocked state.

## Process Monitoring Commands

Use the following commands to monitor processes:

- **top** command—Displays real-time CPU usage statistics on the system. See the [“top Command” section on page 1-10](#).
- **show processes pidin** command—Displays raw output of all processes, including their state.
- **show processes blocked** command—Displays details about reply, send, and mutex blocked processes. See the [“show processes blocked Command” section on page 8-178](#)

You can also use the **monitor processes** and **monitor threads** commands to determine the top processes and threads based on CPU usage.



**Tip**

The top processes command displays almost real-time CPU and memory utilization, and updates several times per minute. The show processes cpu command displays data that has been collected for all process IDs over the past one, five and 15 minute intervals. Both methods provide valuable information.

## Monitoring CPU Usage and Using Syslog Messages

Wdsysmon continuously monitors the system to ensure that no high priority thread is waiting and provides a procedure to recover from high-priority CPU usage. When a process is determined to be a CPU-hog, it is terminated and a coredump of the process is captured and stored on the configured device (exception choice) to aid debugging. For information on troubleshooting high CPU usage, see the [“Troubleshooting High CPU Utilization and Process Timeouts” section on page 8-185](#).

When wdsysmon detects a CPU-hog condition a syslog message is generated. Follow the recommended action for the following syslog messages:

**Message: %HA-HA_WD-6-CPU_HOG_1 CPU hog: cpu [dec]'s sched count is [dec].**

```
RP/0/RP0/CPU0:Dec 22 16:16:34.791 : wdsysmon[331]: %HA-HA_WD-6-CPU_HOG_1 : CPU hog: cpu
1's sched count is 0.
```

Wdsysmon has detected a CPU starvation situation. This is a potentially high priority process spinning in a tight loop. The ‘sched count’ is the number of times the wdsysmon ticker thread has been scheduled since the last time the wdsysmon watcher thread ran.

Check the system status, including the saved log for evidence of a high priority CPU hog. See the [“Troubleshooting High CPU Utilization and Process Timeouts” section on page 8-185](#) for information on checking system status.

**Message: %HA-HA_WD-6-CPU_HOG_2 CPU hog: cpu [dec]'s ticker last ran [dec].[dec] seconds ago.**

```
RP/0/RP0/CPU0:Dec 22 16:16:34.791 : wdsysmon[331]: %HA-HA_WD-6-CPU_HOG_2 : CPU hog: cpu
1's ticker last ran 3.965 seconds ago.
```

Wdsysmon has detected a CPU starvation situation. This is a potentially high priority process spinning in a tight loop.

Check the system status, including the saved log for evidence of a high priority CPU hog. See the [“Troubleshooting High CPU Utilization and Process Timeouts” section on page 8-185](#) for information on checking system status.

**Message: %HA-HA_WD-6-CPU_HOG_3 Rolling average of scheduling times: [dec].[dec].**

```
RP/0/RP0/CPU0:Dec 22 16:16:34.791 : wdsysmon[331]: %HA-HA_WD-6-CPU_HOG_3 : Rolling average
of scheduling times: 0.201.
```

Wdsysmon has detected a CPU starvation situation. This is a potentially high priority process spinning in a tight loop. A high value for the rolling average indicates that a periodic process is not being scheduled.

Check the system status, including the saved log for evidence of a high priority CPU hog. See the [“Troubleshooting High CPU Utilization and Process Timeouts” section on page 8-185](#) for information on checking system status.

**Message: %HA-HA_WD-6-CPU_HOG_4 Process [chars] pid [dec] tid [dec] prio [dec] using [dec]% is the top user of CPU**

```
RP/0/RP0/CPU0:Dec 22 16:16:35.813 : wdsysmon[331]: %HA-HA_WD-6-CPU_HOG_4 : Process wd_test
pid 409794 tid 2 prio 14 using 99% is the top user of CPU.
```

This message is displayed after the CPU hog detector trips. It shows the percentage of CPU used by the busiest thread in the top user of CPU. See the [“Troubleshooting High CPU Utilization and Process Timeouts” section on page 8-185](#) for information on checking system status.

The **show watchdog trace** command displays additional information about the potential CPU hog. If there is a persistent CPU hog (a hog that lasts for more than 30 seconds) the node will be reset. There will be a log such as the following just before the reset:

```
RP/0/RP0/CPU0:Dec 20 10:36:08.990 : wdsysmon[367]: %HA-HA_WD-1-CURRENT_STATE : Persistent
Hog detected for more than 30 seconds
```

If the hog is persistent and the node is reset, contact Cisco Technical Support. For contact information for Cisco Technical Support, see the [“Obtaining Documentation and Submitting a Service Request” section on page viii](#) in the [Preface](#). Copy the error message exactly as it appears on the console or in the system log and provide the representative with the gathered information.

**Note**

For more information on wdsysmon and memory thresholds, see the [“Watchdog System Monitor” section on page 9-197](#) in [Chapter 9, “Troubleshooting Memory.”](#)

## Troubleshooting High CPU Utilization and Process Timeouts

This section describes the troubleshooting of common problems that can occur due to high CPU utilization, and in some cases causing process timeouts. It includes the following topics:

- [General Guidelines for Troubleshooting CPU Utilization Problems, page 8-185](#)
- [Troubleshooting a Process Block, page 8-188](#)
- [Troubleshooting a Process Crash on Line Cards, page 8-192](#)
- [Troubleshooting a Memory Leak, page 8-193](#)
- [Troubleshooting a Hardware Failure, page 8-194](#)
- [Troubleshooting SNMP Timeouts, page 8-194](#)
- [Troubleshooting Communication Among Multiple Processes, page 8-194](#)

### General Guidelines for Troubleshooting CPU Utilization Problems

Optimal CPU utilization is vital for the routers to function properly. In general, the following cases can cause high CPU utilization:

- Normal conditions—One or more processes might be using a large percentage (or all) of the available CPU due to the following reasons:
  - Routing table convergence calculations (until the routing table converges)
  - SNMP polling
  - Any query that requires a large amount of CPU
  - Communication among multiple processes
- Abnormal conditions—A process might be using excessive CPU due to the following reasons:
  - Process (thread) loop
  - Memory leak
  - Process blocking due to bug or hardware problem that causes other process(es) waiting for a reply (loop)

There is no single definition of “high CPU utilization.” Utilization depends on many factors, including the number of clients served and the current configuration on the router. The following example illustrates one approach to troubleshooting utilization. (Details of the commands are provided in the sections that follow.)

**Example:**

You run the **top processes** command. (It shows the top ten processes in terms of CPU usage.)

From the output of the command, you notice that the top two processes use more memory than the next eight. It is possible that this indicates a problem.

You continue by considering the context of this CPU usage. You notice that the top process is OSPF, so you run commands to show whether there are packet drops occurring on the connections that use OSPF. If there are OSPF packet drops, there might be a problem with OSPF that needs attention.

You continue by troubleshooting OSPF. After correcting the OSPF problem, you can rerun the **top processes** command to verify that the CPU usage by the OSPF has been reduced.

## Using show process and top processes Commands

To troubleshoot high CPU utilization due to one of the above reasons, use the following commands:

- **show processes cpu | exclude 0% 0% 0%**—Displays all processes currently using the CPU. The sample output displays high percentages. Run this command multiple times.
- **top processes**—Displays the processes with the most CPU usage.

The top processes command displays almost real-time CPU and memory utilization, and updates several times per minute. The show processes cpu command displays data that has been collected for all process IDs over the past one, five and 15 minute intervals. Both methods provide valuable information.

- **show processes blocked location** *location-id* (Run this command multiple times)
- **show process process_name location** *location-id*
- **follow process process-id location** *location-id*



### Caution

If your system is running Release 3.8.0, 3.8.1, 3.8.2, or 3.9.0 software, you should *not* run the **follow process** and **follow job** commands, because these can cause a kernel crash at the target node. Therefore, for these software releases, you should use other available commands for troubleshooting and call Cisco Technical Support if the problem is not resolved. (This crash behavior does not occur for releases other than the ones listed.)

The following example shows the processing using the CPU.

```
RP/0/RP0/CPU0:router# show processes cpu | exclude 0% 0% 0%

CPU utilization for one minute: 100%; five minutes: 100%; fifteen minutes: 100%

PID      1Min    5Min    15Min Process
24615    98%     97%     97% syslog_dev  <---!!!
65647     1%      1%      1% bfd_agent

RP/0/RP0/CPU0:CIPC2-VAN#
RP/0/RP0/CPU0:CIPC2-VAN#show process block loc 0/0/cpu0
Jid      Pid Tid      Name State Blocked-on
 54      8202 1          ksh Reply 8199 devc-ser8250
 51      20502 2          attachd Reply 20500 eth_server
 51      20502 3          attachd Reply 8204 mqueue
 72      20503 6          qnet Reply 20500 eth_server
 72      20503 7          qnet Reply 20500 eth_server
 72      20503 8          qnet Reply 20500 eth_server
 72      20503 9          qnet Reply 20500 eth_server
 52      20507 1          ksh-aux Reply 8199 devc-ser8250
 50      20508 2          attach_server Reply 8204 mqueue
 216     24610 1          reddrv_listener Reply 20500 eth_server
 246     90234 1          spa_xge_v2 Reply 24615 syslog_dev  <---!!!
 246     90234 5          spa_xge_v2 Mutex 90234-01 #1
```

```
RP/0/RP0/CPU0:CIPC2-VAN#show process block loc 0/0/cpu0
Jid      Pid Tid      Name State Blocked-on
 54      8202 1      ksh Reply 8199 devc-ser8250
 51      20502 2      attachd Reply 20500 eth_server
 51      20502 3      attachd Reply 8204 mqueue
 72      20503 6      qnet Reply 20500 eth_server
 72      20503 7      qnet Reply 20500 eth_server
 72      20503 8      qnet Reply 20500 eth_server
 72      20503 9      qnet Reply 20500 eth_server
 52      20507 1      ksh-aux Reply 8199 devc-ser8250
 50      20508 2      attach_server Reply 8204 mqueue
216      24610 1      reddrv_listener Reply 20500 eth_server
246      90234 1      spa_xge_v2 Reply 24615 syslog_dev <--still blocking!!!
246      90234 5      spa_xge_v2 Mutex 90234-01 #1
RP/0/RP0/CPU0:CIPC2-VAN#

RP/0/RP0/CPU0:CIPC2-VAN#show process syslog_dev loc 0/0/cpu0
Tue Sep 11 17:22:51.182 UTC
      Job Id: 262
      PID: 24615
      Executable path: /bootflash/hfr-base-3.4.1/bin/syslog_dev
      Instance #: 1
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
      Max. spawns per minute: 12
      Last started: Fri Jun 22 14:15:01 2007
      Process state: Run
      Package state: Normal
      core: TEXT SHARED MEM MAIN MEM
      Max. core: 0
      Level: 40.90
      MaintModeProc: ON
      startup_path: /pkg/startup/syslog_dev.startup
      Ready: 0.999s
      Process cpu time: 1283052.366 user, 0.291 kernel, 1283052.657 total
JID      TID Stack pri state      HR:MM:SS:MSEC NAME
262      1      12K 10 Ready      1549:26:59:0925 syslog_dev <---take look at cpu time
spending for this process!!!
```

The following example shows the processes with the most CPU usage.

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

Computing times...

247 processes; 930 threads; 4804 channels, 6683 fds

CPU states: 98.5% idle, 0.6% user, 0.8% kernel

Memory: 4096M total, 3095M avail, page size 4K

JID	TIDS	Chans	FDs	Tmrs	MEM	HH:MM:SS	CPU	NAME
1	33	250	197	1	0	437:50:44	0.82%	procnto-600-smp-cisco-i
333	9	32	21	16	1M	0:11:28	0.26%	sysdb_svr_admin
180	21	132	40	11	6M	0:36:37	0.16%	gsp
332	7	161	19	11	1M	0:10:33	0.12%	sysdb_mc
376	7	31	62	13	6M	0:06:49	0.04%	mpls_ldp
159	1	5	14	2	756K	0:00:48	0.04%	envmon_mon
344	5	6	47	2	1M	0:01:39	0.02%	top_procs
341	35	26	62	6	728K	0:01:02	0.02%	tcp
276	3	9	15	2	548K	0:00:06	0.00%	oir_daemon
62	1	6	9	1	204K	0:00:07	0.00%	i2c_server


## Troubleshooting a Process Block


To troubleshoot a blocked process, perform the following procedure.

### SUMMARY STEPS

1. **show processes blocked location** *node-id*
2. **follow job** *job-id* **location** *node-id*
3. **process restart** *job-id* **location** *node-id*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show processes blocked location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show processes blocked location 0/0/cpu0	<p>Use the <b>show processes blocked</b> command several times (three times at 5 second intervals) and compare the output to determine if any processes are blocked for a long period of time. The process can be blocked continuously or for a few seconds. A process is blocked while it is waiting for a response from another process.</p> <ul style="list-style-type: none"> <li>• The Name column shows the name of the blocked process.</li> <li>• The Blocked-on column shows the name and process ID of the blocked process.</li> <li>• If the State column is Mutex, a thread in the process waits for another thread. In this case, the Blocked-on column shows the process ID and thread ID instead of the process ID.</li> <li>• A blocked process can be blocked by another process. If a process is being blocked by another process, you need to track the chain of blocking and find the root of blocking processes. Proceed to <a href="#">Step 2</a> to track the chain of blocking.</li> </ul>
Step 2	<b>follow job</b> <i>job-id</i> <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# follow job 24615 location 0/0/cpu0	<p>Tracks the root process blocking other processes. The <b>follow</b> command shows what part of the code the process is periodically executed.</p> <div>  <p><b>Caution</b> If your system is running Release 3.8.0, 3.8.1, 3.8.2, or 3.9.0 software, you should <i>not</i> run the <b>follow process</b> and <b>follow job</b> commands, because these can cause a kernel crash at the target node. Therefore, for these software releases, you should use other available commands for troubleshooting and call Cisco Technical Support if the problem is not resolved. (This crash behavior does not occur for releases other than the ones listed.)</p> </div>

Command or Action	Purpose
<b>Step 3</b> <b>process restart</b> <i>job-id</i> <i>location</i> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# process restart 234 location 0/1/cpu0	Restarts the process. If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.  Collect the following information for Cisco Technical Support: <ul style="list-style-type: none"> <li>• <b>show processes blocked</b> <i>location</i> <i>node-id</i> command output</li> <li>• <b>follow job</b> <i>job-id</i> <i>location</i> <i>node-id</i> command output</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p><b>Caution</b> If your system is running Release 3.8.0, 3.8.1, 3.8.2, or 3.9.0 software, you should <i>not</i> run the <b>follow process</b> and <b>follow job</b> commands, because these can cause a kernel crash at the target node. Therefore, for these software releases, you should use other available commands for troubleshooting and call Cisco Technical Support if the problem is not resolved. (This crash behavior does not occur for releases other than the ones listed.)</p> </div> <ul style="list-style-type: none"> <li>• <b>show version</b> command output</li> <li>• <b>show dll</b> command output</li> <li>• <b>show configuration</b> command output</li> <li>• <b>show logging</b> command output</li> <li>• content of the file: disk0:/wdsysmon_debug/debug_env. <i>number</i> (if it exists)</li> </ul>

The following example shows the details for a blocked process for CPU usage:

```
RP/0/RP0/CPU0:router# show processes cpu location 0/0/cpu0 | exc 0%      0%      0%
```

CPU utilization for one minute: 100%; five minutes: 100%; fifteen minutes: 100%

```
PID      1Min      5Min      15Min Process
24615    98%       97%      97% syslog_dev <---!!!
65647     1%        1%       1%  bfd_agent
```

The following example shows the details for active processes from a designated node:

```
RP/0/RP0/CPU0:router# show processes blocked location 0/0/cpu0
```

```
Jid      Pid Tid      Name State Blocked-on
54        8202 1      ksh Reply 8199 devc-ser8250
51        20502 2     attachd Reply 20500 eth_server
51        20502 3     attachd Reply 8204 mqueue
72        20503 6       qnet Reply 20500 eth_server
72        20503 7       qnet Reply 20500 eth_server
72        20503 8       qnet Reply 20500 eth_server
```

```

72      20503      9              qnet Reply      20500  eth_server
52      20507      1              ksh-aux Reply      8199  devc-ser8250
50      20508      2      attach_server Reply      8204  mqueue
216     24610      1      reddrv_listener Reply      20500  eth_server
246     90234      1              spa_xge_v2 Reply      24615  syslog_dev  <--!!!
246     90234      5              spa_xge_v2 Mutex      90234-01 #1

```

The following example gathers information about the dump core on the blocked process:

```
RP/0/RP0/CPU0:router# follow process 24615 location 0/0/cpu0
```

```
Tue Sep 11 17:21:26.205 UTC
```

```
Attaching to process pid = 24615 (pkg/bin/syslog_dev)
No tid specified, following all threads
```

```
DLL Loaded by this process
```

```
-----
```

DLL path	Text addr.	Text size	Data addr.	Data size	Version
/pkg/lib/libsysmgr.dll	0xfc124000	0x00010f9c	0xfc087a28	0x000005cc	0
/pkg/lib/libcerrno.dll	0xfc135000	0x00002f9c	0xfc1126ac	0x00000128	0
/pkg/lib/libcerr_dll_tbl.dll	0xfc138000	0x000049e0	0xfc1127d4	0x00000148	0
/pkg/lib/libltrace.dll	0xfc13d000	0x00008a60	0xfc11291c	0x000002c8	0
/pkg/lib/libinfra.dll	0xfc146000	0x00034e60	0xfc17b000	0x00002000	0
/pkg/lib/cerrno/libinfra_error.dll	0xfc1141dc	0x00000cd8	0xfc112be4	0x000000a8	0
/pkg/lib/libbios.dll	0xfc17d000	0x0002cc34	0xfc1aa000	0x00002000	0
/pkg/lib/cerrno/libevent_manager_error.dll	0xfc1ac000	0x00000e88	0xfc112c8c	0x00000088	0
/pkg/lib/libc.dll	0xfc1ad000	0x0007b118	0xfc229000	0x00002000	0
/pkg/lib/libplatform.dll	0xfc23f000	0x0000c738	0xfc24c000	0x00002000	0
/pkg/lib/libnodeid.dll	0xfc24e000	0x0000a730	0xfc23a3f8	0x00000248	0
/pkg/lib/libdebug.dll	0xfc25c000	0x00010038	0xfc23a7cc	0x00000550	0
/pkg/lib/cerrno/libdebug_error.dll	0xfc26c038	0x00000db0	0xfc23ad1c	0x000000e8	0
/pkg/lib/lib_procfs_util.dll	0xfc26d000	0x00004fb8	0xfc272000	0x000002a8	0
/pkg/lib/libsyslog.dll	0xfc28f000	0x0000564c	0xfc2724c0	0x00000328	0
/pkg/lib/libbackplane.dll	0xfc295000	0x000013f0	0xfc2727e8	0x000000a8	0
/pkg/lib/cerrno/libsysmgr_error.dll	0xfc4c9000	0x00000f94	0xfc2fba04	0x00000088	0
/pkg/lib/libsysdb.dll	0xfc4d9000	0x0004a000	0xfc523000	0x00001000	0
/pkg/lib/cerrno/libsysdb_error_v1v2.dll	0xfc524000	0x00002000	0xfc526000	0x00001000	0
/pkg/lib/cerrno/libsysdb_error_v2only.dll	0xfc527000	0x00003000	0xfc52a000	0x00001000	0
/pkg/lib/cerrno/libsysdb_error_callback.dll	0xfc52b000	0x00002000	0xfc52d000	0x00001000	0
/pkg/lib/cerrno/libsysdb_error_distrib.dll	0xfc52e000	0x00002000	0xfc530000	0x00001000	0
/pkg/lib/libsysdbutils.dll	0xfc531000	0x0000d000	0xfc53e000	0x00001000	0

```
Iteration 1 of 5
```

```
-----
```

```
Current process = "pkg/bin/syslog_dev", PID = 24615 TID = 1
```

```

trace_back: #0 0xfc1f6044 [strlen]
trace_back: #1 0x482002c8 [<N/A>]
trace_back: #2 0x48200504 [<N/A>]
trace_back: #3 0xfc1e4408 [_resmgr_io_handler]
trace_back: #4 0xfc1e40b0 [_resmgr_handler]
trace_back: #5 0xfc15aa28 [_eventmgr_resmgr_handler]
trace_back: #6 0xfc159e04 [_event_message_handler]
trace_back: #7 0xfc159d54 [_event_message_handler]
trace_back: #8 0xfc1575a4 [event_dispatch]
trace_back: #9 0x482007bc [<N/A>]

```



ENDOFSTACKTRACE

Iteration 2 of 5

-----

Current process = "pkg/bin/syslog_dev", PID = 24615 TID = 1

trace_back: #0 0xfc1f6130 [strncat]  
trace_back: #1 0x482002a4 [<N/A>]  
trace_back: #2 0x48200504 [<N/A>]  
trace_back: #3 0xfc1e4408 [_resmgr_io_handler]  
trace_back: #4 0xfc1e40b0 [_resmgr_handler]  
trace_back: #5 0xfc15aa28 [_eventmgr_resmgr_handler]  
trace_back: #6 0xfc159e04 [_event_message_handler]  
trace_back: #7 0xfc159d54 [_event_message_handler]  
trace_back: #8 0xfc1575a4 [event_dispatch]  
trace_back: #9 0x482007bc [<N/A>]

ENDOFSTACKTRACE

Iteration 3 of 5

-----

Current process = "pkg/bin/syslog_dev", PID = 24615 TID = 1

trace_back: #0 0xfc1f6044 [strlen]  
trace_back: #1 0x482002c8 [<N/A>]  
trace_back: #2 0x48200504 [<N/A>]  
trace_back: #3 0xfc1e4408 [_resmgr_io_handler]  
trace_back: #4 0xfc1e40b0 [_resmgr_handler]  
trace_back: #5 0xfc15aa28 [_eventmgr_resmgr_handler]  
trace_back: #6 0xfc159e04 [_event_message_handler]  
trace_back: #7 0xfc159d54 [_event_message_handler]  
trace_back: #8 0xfc1575a4 [event_dispatch]  
trace_back: #9 0x482007bc [<N/A>]

ENDOFSTACKTRACE

Iteration 4 of 5

-----

Current process = "pkg/bin/syslog_dev", PID = 24615 TID = 1

trace_back: #0 0xfc1f6044 [strlen]  
trace_back: #1 0x482002c8 [<N/A>]  
trace_back: #2 0x48200504 [<N/A>]  
trace_back: #3 0xfc1e4408 [_resmgr_io_handler]  
trace_back: #4 0xfc1e40b0 [_resmgr_handler]  
trace_back: #5 0xfc15aa28 [_eventmgr_resmgr_handler]  
trace_back: #6 0xfc159e04 [_event_message_handler]  
trace_back: #7 0xfc159d54 [_event_message_handler]  
trace_back: #8 0xfc1575a4 [event_dispatch]  
trace_back: #9 0x482007bc [<N/A>]

ENDOFSTACKTRACE

Iteration 5 of 5

-----

```

Current process = "pkg/bin/syslog_dev", PID = 24615 TID = 1

trace_back: #0 0xfc1f6130 [strncat]
trace_back: #1 0x482002a4 [<N/A>]
trace_back: #2 0x48200504 [<N/A>]
trace_back: #3 0xfc1e4408 [_resmgr_io_handler]
trace_back: #4 0xfc1e40b0 [_resmgr_handler]
trace_back: #5 0xfc15aa28 [_eventmgr_resmgr_handler]
trace_back: #6 0xfc159e04 [_event_message_handler]
trace_back: #7 0xfc159d54 [_event_message_handler]
trace_back: #8 0xfc1575a4 [event_dispatch]
trace_back: #9 0x482007bc [<N/A>]

ENDOFSTACKTRACE

```

## Troubleshooting a Process Crash on Line Cards

To troubleshoot a process crash on the line card, perform the following steps.

### SUMMARY STEPS

1. Identify the process that crashed (PI or PD) from the crash log. In either case, the stack traces obtained from the crash needs to be decoded to identify the location in the code where the process crashed.
2. **show install active**
3. **show version**
4. **show log**
5. **show exception**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show install active</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show install active location 0/0/CPU0 Node 0/0/CPU0 [RP] [SDR: Owner] Boot Device: mem: Boot Image: /c12k-os-mbi-3.7.0.26I/mbiprp-rp.vm Active Packages: mem:c12k-mpls-3.7.0.26I mem:c12k-mini-3.7.0.26I	Use the <b>show install active</b> command to collect information about the list of installed software for each node.
Step 2	<b>show version</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show version   begin 0/0/CPU0	Gives the workspace (directory) and the build server where image was built.

	Command or Action	Purpose
Step 3	<b>show log</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show log	Provides a background on what was going on at the time of the crash. You can find syslog messages from the dumper process at the time of the crash. This provides a list of dynamic libraries which were loaded by the process and the addresses where they were mapped. This is required to decode the program counters in the stack trace which are a part of a DLL. Also, the location where the core dump has been saved is available.  The core dump is the .Z file mentioned in the log.
Step 4	<b>show exception</b>	If you don't have the log, use the <b>show exception</b> command to find out where it is saved.  If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Troubleshooting a Memory Leak

To troubleshoot a memory leak, use the **show processes {files} [job-id] [detail]** command to display detailed information about open files and open communication channels. The *job-id* argument displays the job identifier information for only the associated process instance.

The following example shows output from the **show processes** command with the **files** and **detail** keywords:

```
RP/0/RP0/CPU0:router# show processes files 351 detail
```

```
Sun Jan 21 04:35:18.451 EDT
Jid: 351          Total open files: 352          Name: tacacsd
```

```
-----
File Descriptor  Process Name
-----
0                pid: 1
1                pid: 1
2                syslog_dev
3                dllmgr
4                pid: 1
5                pid: 1
6                sysdb_svr_local
7                sysdb_svr_local
8                sysmgr
9                sysdb_svr_local
10               sysdb_svr_local
11               sysdb_mc
12               sysdb_svr_local
13               sysdb_mc
14               sysdb_svr_local
15               sysdb_mc
16               sysdb_mc
17               sysdb_mc
18               pid: 1
19               tcp
20               pid: 1
21               tcp
```

```

22          tcp
23          tcp
24          tcp
25          tcp
26          tcp
27          tcp
28          tcp
29          tcp
30          tcp
31          tcp
32          tcp
33          tcp
34          tcp
35          tcp
36          tcp
37          tcp
38          tcp
39          tcp
40          tcp
41          tcp

```

## Troubleshooting a Hardware Failure

Hardware failure can have a major impact on the normal operation of CPU. If a problem is detected, messages can be obtained from the syslog or you can get a node name from the output of the **show processes** command with the **blocked** keyword.

## Troubleshooting SNMP Timeouts

This section explains how to troubleshoot a typical SNMP timeout scenario.

The service provider typically initiates an SNMP query by means of an SNMP server in the network operations center. When you set up an SNMP query on an SNMP server, you set the parameters of the query, including a timer. If the timer expires before the server receives the query results, this means the query has timed out. If the requested SNMP query involves a large amount of data from the Cisco CRS-1 (and this is generally true for SNMP queries), the Cisco CRS-1 might experience very high CPU utilization as it searches for the data. The Cisco CRS-1 might not be able to complete the query and data transfer before the timer on the SNMP server expires.

To correct a problem with SNMP timeouts, set the timer on the SNMP server to a higher value.

Process timeouts can also occur if communication among multiple process causes high CPU utilization. For information about this scenario, see the [“Troubleshooting Communication Among Multiple Processes”](#) section on page 8-194.

## Troubleshooting Communication Among Multiple Processes

If communication among multiple processes is causing high CPU utilization, you must stop the request process (for example, Simple Network Management Protocol [SNMP], Internet Control Message Protocol [ICMP], or TCP).

To check the communication blocks among multiple processes, use the **show processes** command. Use the **blocked** keyword (multiple times) to display details about the blocked process. Use the **cpu** keyword (multiple times) to display the CPU usage for each process.

The following example shows the details about the blocked processes from the **show processes** command with the **blocked** keyword:

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

Jid	Pid	Tid	Name	State	Blocked-on
65546	8202	1	ksh	Reply	8200 devc-conaux
51	36889	2	attachd	Reply	32790 eth_server
51	36889	3	attachd	Reply	12301 mqueue
74	36892	5	qnet	Reply	32790 eth_server
74	36892	6	qnet	Reply	32790 eth_server
74	36892	7	qnet	Reply	32790 eth_server
74	36892	8	qnet	Reply	32790 eth_server
50	36898	2	attach_server	Reply	12301 mqueue
361	118859	1	tftp_server	Reply	12301 mqueue
259	139360	2	locald	Reply	233685 tacacsd
261	155823	2	lpts_fm	Reply	139336 lpts_pa
65717	51572917	1	exec	Reply	1 kernel
370	233689	1	udp_snmpd	Reply	155811 udp
65759	51589343	1	more	Reply	12299 pipe
65774	51589358	1	show_processes	Reply	1 kernel

The following example shows the CPU usage for each process from the **show processes** command with the **cpu** keyword:

```
RP/0/RP0/CPU0:router# show processes cpu | exclude 0% 0% 0%
```

PID	1Min	5Min	15Min	Process
1	3%	1%	1%	kernel
32790	4%	4%	3%	eth_server
36892	1%	0%	0%	qnet
114740	9%	8%	8%	sysdb_svr_shared <--!!!
118855	1%	0%	0%	netio
118857	11%	10%	10%	gsp <--!!!
118862	7%	6%	6%	sysdb_mc
155799	1%	1%	1%	ipv4_rib
155802	2%	2%	2%	ipv4_arm
233707	1%	1%	1%	bgp

## Troubleshooting a Process Restart

A process restart does not typically cause a major problem for the network, nor does it typically cause a loss of traffic. However, it can be helpful to troubleshoot the event, to find out *why* the process crashed, and whether you need to consider taking any action to locate or correct a problem.

To troubleshoot a process restart, contact Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the [Preface](#).

Collect the following information before contacting Cisco Technical Support.

- **show context all location all** command output
- **show version** command output
- **show dll** command output
- **show log** command output
- Collect core dumps. See the [“show context Command”](#) section on page 1-11 for information on how to collect core dumps.





## CHAPTER 9

# Troubleshooting Memory

---

Troubleshooting memory requires determining if there is a memory problem, what type of memory problem it is, and how to resolve the problem.

This chapter contains the following sections:

- [Watchdog System Monitor, page 9-197](#)
- [Troubleshooting Global Memory, page 9-202](#)
- [Troubleshooting Process Memory, page 9-203](#)

## Watchdog System Monitor

Watchdog system monitor (wdsysmon) is part of the high-availability (HA) infrastructure of the Cisco IOS XR software. Wdsysmon runs on the distributed route processor (DRP) and line cards (LCs) with the primary goal of monitoring the system for problem conditions and attempting to recover from them. Wdsysmon monitors the processes on each node for memory and CPU usage, deadlocks, and event monitor conditions, as well as disk usage. If thresholds are crossed, an appropriate syslog message is generated. The information is collected in `disk0:/wdsysmon_debug`. Recovery actions are taken when memory of CPU hog or a deadlock condition is detected, whereby the process responsible for the condition is terminated. Wdsysmon also keeps historical data on processes and posts this information to a fault detector dynamic link library (DLL), which can then be queried by manageability applications.

## Memory Monitoring

The wdsysmon memory-hog detection algorithm checks the memory state of each node in regular intervals (every 1/10th of a second). It defines four node state thresholds:

- Normal
- Minor
- Severe
- Critical



### Note

---

Processes can declare a hard memory limit in their startup file with the **memory_limit** keyword.

---

The definition of the node state thresholds depends on the size of the physical memory. For instance, on a node with 2 GB of physical memory, the memory state is considered NORMAL as long the free memory is greater than 48 MB.

If a memory threshold is crossed, wdsysmon immediately checks if a process has exceeded its memory limit. All such processes are stopped after a debug script runs on the process identifier (PID) to collect detailed information on the memory hog. If memory usage is still high after this step, wdsysmon sends notifications to registered clients. Clients can then take preventive and recovery actions.

The memory state can be verified using the **show watchdog memory-state location node-id** command. The following example shows node 0/RP0/CPU0 as in the normal memory state.

```
RP/0/RP0/CPU0:router# show watchdog memory-state location 0/rp0/cpu0
```

```
Memory information:
  Physical Memory: 4096      MB
  Free Memory:    3485.671 MB
  Memory State:   Normal
```

If the memory state is changing from normal to minor use the **show processes memory [job-id] location node-id** command to list top memory users and identify possible memory leaks. After top memory users have been identified, use the memory usage analyzer to discover the processes causing a memory leak. See the [“Memory Usage Analyzer” section on page 9-202](#). Your technical representative should now be involved to collect the appropriate data and take the corresponding actions such as process restart.

Wdsysmon has a procedure to recover from memory-depletion conditions. When wdsysmon determines that the state of a node is severe, it attempts to find a process, or set of processes, that have likely leaked memory leading to the depletion condition. The process or set of processes are stopped to recover the memory. This situation should be avoided by regularly checking the watchdog memory state.

## Configuring and Displaying Memory Thresholds

Memory thresholds can be configured. Threshold values can be applied to all cards, or unique threshold settings can be applied to specific cards. If the local threshold settings are removed, the local settings return to those set globally. In addition, you can view default and configured thresholds.

[Table 9-1](#) provides the recommended memory threshold value calculations if the minor threshold is set to 20 percent, the severe threshold is set to 10 percent, and the critical threshold is set to 5 percent.

**Table 9-1 Recommended Memory Threshold Values**

Total Available Memory (MB)	Minor Threshold (20 percent of available memory)	Severe Threshold (10 percent of available memory)	Critical Threshold (5 percent of available memory)
128	25.6	12.8	6.4
256	51.2	25.6	12.8
512	102.4	51.2	25.6
1024	204.8	102.4	51.2
2048	409.6	204.8	102.4
4096	819.2	409.6	204.8

To identify, configure, and display memory thresholds, perform the following procedure.



## SUMMARY STEPS

1. **configure**
2. **watchdog memory threshold** [*location node-id*] **minor** *percentage-memory-available* **severe** *percentage-memory-available* **critical** *percentage-memory-available*
3. **end**  
or  
**commit**
4. **exit**
5. **show watchdog** [*memory-state* | *threshold memory {configured | default}*] [*location node-id*]
6. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>  <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	<b>watchdog threshold memory</b> [ <i>location node-id</i> ] <b>minor</b> <i>percentage-memory-available</i> <b>severe</b> <i>percentage-memory-available</i> <b>critical</b> <i>percentage-memory-available</i>  <b>Example:</b> RP/0/RP0/CPU0:router(config)# watchdog threshold memory location 0/RP0/CPU0 minor 30 severe 20 critical 10	Configures the value of memory available for each alarm threshold. For example, if the minor alarm threshold is set to 30 percent, the severe alarm threshold is set to 20 percent, and the critical alarm threshold is set to 10 percent, the node goes into a minor memory alarm when the amount of memory available falls below 30 percent of the total memory on the card. In other words, this alarm occurs when 70 percent of the available memory is in use. The severe memory alarm activates when the amount of memory available falls below 20 percent, and the critical memory alarm activates when the amount of memory available falls below 10 percent.

	Command or Action	Purpose
Step 3	<b>end</b> or <b>commit</b>  <b>Example:</b> RP/0/RP0/CPU0:router(config)# end or RP/0/RP0/CPU0:router(config)# commit	Saves configuration changes. <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes:  Uncommitted changes found, commit them before exiting(yes/no/cancel)?  [cancel]: <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
Step 4	<b>exit</b>  <b>Example:</b> RP/0/RP0/CPU0:router(config)# exit	Exits global configuration mode and enters EXEC mode.
Step 5	<b>show watchdog</b> [ <b>memory-state</b>   <b>threshold memory</b> { <b>configured</b>   <b>defaults</b> }] [ <b>location</b> <i>node-id</i> ]  <b>Example:</b> RP/0/RP0/CPU0:router# show watchdog threshold memory configured location 0/RP0/CPU0	Displays information about the threshold memory in the specified locations, either as configured by the user or for the default settings.
Step 6	Contact Cisco Technical Support.	If the problem has not been determined and is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The **watchdog memory threshold** enables you to set the memory thresholds.

```
RP/0/RP0/CPU0:router(config)# watchdog threshold memory location 0/RP0/CPU0 minor 30
severe 20 critical 10
```

The **show watchdog threshold memory default** command enables you to display the default memory thresholds.

```
RP/0/RP0/CPU0:router# show watchdog threshold memory defaults location all
```

```
[K---- node0_RP1_CPU0 ---
Default memory thresholds:
Minor:    409      MB
Severe:   204      MB
```

```

Critical: 102.399 MB
Memory information:
  Physical Memory: 2048      MB
  Free Memory:    1236.296 MB
  Memory State:   Normal
---- node0_3_SP ---
Default memory thresholds:
Minor:    25      MB
Severe:   12      MB
Critical: 6.399 MB
Memory information:
  Physical Memory: 128      MB
  Free Memory:    41.187 MB
  Memory State:   Normal
---- node0_0_SP ---
Default memory thresholds:
Minor:    25      MB
Severe:   12      MB
Critical: 6.399 MB
[KMemory information:
  Physical Memory: 128      MB
  Free Memory:    40.683 MB
  Memory State:   Normal
---- node0_SM0_SP ---
Default memory thresholds:
Minor:    25      MB
Severe:   12      MB
Critical: 6.399 MB
Memory information:
  Physical Memory: 128      MB
  Free Memory:    34.394 MB
  Memory State:   Normal
---- node0_0_CPU0 ---
Default memory thresholds:
Minor:    204     MB
Severe:   102     MB
Critical: 51.199 MB
Memory information:
  Physical Memory: 1024     MB
  Free Memory:    463.304 MB
  Memory State:   Normal
---- node0_RP0_CPU0 ---
[K Default memory thresholds:
Minor:    819     MB
Severe:   409     MB
Critical: 204.799 MB
Memory information:
  Physical Memory: 4096     MB
  Free Memory:    33

```

## Setting Timeout for Persistent CPU Hogs

If wdsysmon detects a CPU hog on the card, it resets the node after 30 seconds. This default timeout value can be reset if required using the **watchdog monitor cpu-hog persistent timeout** command.

## Memory Usage Analyzer

The memory usage analyzer tool records brief details about the heap memory usage of all processes on the router at different moments in time and compares the results. This makes it very useful for detecting patterns of memory usage during events such as restarting processes or configuring interfaces. It is also useful for troubleshooting memory leaks.

When the memory usage analyzer tool is instructed to take a snapshot, it saves output similar to the **show memory heap summary** command output for each process running on the router to a file. When instructed to show a report, the files are read and the differences between the memory values are displayed.

The memory usage analyzer tool uses the following commands in sequence:

1. **show memory compare start** command—This command takes an initial snapshot of the process memory usage.
2. **show memory compare end** command—This command takes another snapshot of the process memory usage.
3. **show memory compare report** command—This command displays the differences between the memory values.

The command output contains information about each process whose heap memory usage has changed over the test period. It is ordered by the size of the change, starting with the process with the largest increase. To detect memory leaks the memory usage analyzer should be used on a stable system when no configuration changes are taken.

## Troubleshooting Global Memory

To begin troubleshooting a router in a low memory state, get a high-level view of where the memory is being used.

Use the **show memory summary** command to display system memory information.

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

```
Physical Memory: 4096M total  
Application Memory : 3949M (3540M available)  
Image: 17M (bootram: 17M)  
Reserved: 128M, IOMem: 2028M, flashfsys: 0  
Total shared window: 7M
```

The output shows the amount of physical memory installed on the device, the memory available for the system to use (total memory minus image size, reserved, and flashfsys), the image size, the reserved space for packet memory, and the I/O memory used as a backup for packet memory.

If there is not sufficient memory, install more memory. See the Cisco CRS-1 Carrier Routing System documentation at the following URL:

[http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html](http://www.cisco.com/en/US/products/ps5763/tsd_products_support_series_home.html)

# Troubleshooting Process Memory

The Cisco IOS XR Process Placement feature balances application processes between the available route processors (RPs) and distributed route processors (DRPs) on a Cisco CRS-1 system, based on memory usage and other criteria.

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.

## Identifying Process Memory Problems

To identify process memory problems, perform the following procedure.

### SUMMARY STEPS

1. **show watchdog memory-state location** *node-id*
1. **show processes memory** [*job-id*] **location** *node-id*
2. **show memory** *job-id*
3. **show process memory** *job-id*
4. **show memory compare start**
5. **show memory compare end**
6. **show memory compare report**
7. Contact Cisco Technical Support if the problem is not resolved.

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show watchdog memory-state location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show watchdog memory-state location 0/RP0/CPU0	Displays the memory state for the node. If the node is not in the normal state, proceed to <a href="#">Step 2</a> to list top memory users and identify possible memory leaks.
Step 2	<b>show processes memory</b> [ <i>job-id</i> ] <b>location</b> <i>node-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# show process memory location 0/RP0/CPU0	Displays information about the text, data, and stack usage for all active processes on a specified node. The output lists top memory users and identifies possible memory leaks. After top memory users have been identified, note the job ID and use the memory usage analyzer to discover the processes causing a memory leak. See <a href="#">Step 5</a> through <a href="#">Step 7</a> for how to use the memory usage analyzer.

	Command or Action	Purpose
Step 3	<b>show memory <i>job-id</i></b>  <b>Example:</b> RP/0/RP0/CPU0:router# show memory 123	Displays the available physical memory and memory usage information of a specific process.
Step 4	<b>show process memory <i>job-id</i></b>  <b>Example:</b> RP/0/RP0/CPU0:router# show process memory 123	Displays information about the text, data, and stack usage for a specific process.
Step 5	<b>show memory compare start</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show memory compare start	Takes the initial snapshot of heap memory usage for all processes on the router and sends the report to a temporary file named /tmp/memcmp_start.out.
Step 6	<b>show memory compare end</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show memory compare end	Takes the second snapshot of heap memory usage for all processes on the router and sends the report to a temporary file named /tmp/memcmp_end.out. This snapshot is compared with the initial snapshot when displaying the heap memory usage comparison report.
Step 7	<b>show memory compare report</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show memory compare report s	Displays the heap memory comparison report, comparing heap memory usage between the two snapshots of heap memory usage.
Step 8	Contact Cisco Technical Support.	If the problem has not been determined and is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.

## Examples

The **show watchdog memory-state location *node-id*** command allows you to determine the memory state of a specified node.

```
RP/0/RP0/CPU0:router# show watchdog memory-state location 0/rp0/cpu0
```

```
Memory information:
  Physical Memory: 4096      MB
  Free Memory:    3485.671 MB
  Memory State:   Normal
```

The **show process memory [*job-id*] location *node-id*** command allows you to determine the processes with the highest dynamic memory usage. The output of the command is sorted by the Dynamic memory usage.

```
RP/0/RP0/CPU0:router# show processes memory location 0/rp0/cpu0
```

```
JID   Text      Data      Stack     Dynamic   Process
59    65536     32768     57344     38064128  eth_server
164   147456    4096      24576     13217792  fgid_server
289   90112     4096      94208     8437760   parser_server
65554 40960      0         32768     7430144   devb-ata
```

```

181 110592 4096 151552 3350528 gsp
57 28672 0 28672 3284992 dllmgr
335 4096 4096 36864 3194880 sysdb_svr_local
280 53248 4096 20480 2682880 nvgen_server
319 16384 4096 12288 2482176 schema_server
329 81920 4096 40960 2412544 statsd_manager
67 28672 12288 24576 2306048 nvram
360 552960 4096 69632 2232320 wdsysmon
216 98304 4096 65536 1908736 ipv4_rib
336 4096 4096 77824 1806336 sysdb_svr_shared
193 217088 4096 86016 1613824 ifmgr
273 45056 4096 122880 1544192 netio
234 98304 4096 53248 1486848 ipv6_rib
190 36864 4096 36864 1429504 hd_drv
333 53248 4096 65536 1327104 sysdb_mc
175 45056 4096 49152 1277952 fabricq_mgr
379 4096 0 94208 1212416 xmlagent
334 4096 4096 73728 1200128 sysdb_svr_admin
204 40960 4096 24576 262144 ipv4_acl_dispatch
162 12288 4096 12288 262144 ether_caps_partner
375 4096 0 12288 245760 ipsec_simp
196 12288 4096 12288 237568 imaedm_server
123 4096 4096 16384 237568 bgp_policy_reg_agent
222 32768 4096 20480 233472 ipv6_acl_daemon
315 16384 4096 61440 229376 rt_check_mgr
.
.
.

```

The **show memory job-id** command displays the memory available and memory usage information for the process associated with the specified job ID. The command output allows you to see exactly what memory is allocated by the process.

```
RP/0/RP0/CPU0:router# show memory 123
```

```

Physical Memory: 4096M total
Application Memory : 3949M (3540M available)
Image: 17M (bootram: 17M)
Reserved: 128M, IOMem: 2028M, flashfsys: 0
Shared window ipv4_fib: 1M
Shared window infra_ital: 323K
Shared window ifc-mpls: 961K
Shared window ifc-ipv6: 1M
Shared window ifc-ipv4: 1M
Shared window ifc-protomax: 641K
Shared window aib: 203K
Shared window infra_statsd: 3K
Shared window PFI_IFH: 155K
Shared window squid: 2M
Shared window atc_cache: 35K
Total shared window: 7M

pkg/bin/bgp_policy_reg_agent: jid 123
Address      Bytes      What
4817f000     4096      Program Stack (pages not allocated)
48180000     507904    Program Stack (pages not allocated)
481fc000     16384     Program Stack
48200000     16384     Shared Memory
48204000     4096      Program Text or Data
48205000     4096      Program Text or Data
48206000     16384     Allocated Memory
4820a000     16384     Allocated Memory
4820e000     16384     Allocated Memory
48212000     16384     Allocated Memory

```

```

48216000      16384      Allocated Memory
4821a000      16384      Allocated Memory
4821e000      16384      Allocated Memory
48222000      16384      Allocated Memory
60100000       8192      Shared Memory
60102000      36864      Shared Memory
6010b000      102400     Shared Memory
60124000       8192      Shared Memory
.
.
.
fd214000      106496     DLL Text liboradock.dll
fd22e000       4096      DLL Data liboradock.dll
fd241000       49152     DLL Text librasf.dll
Total Allocated Memory: 131072
Total Shared Memory: 978944

```

The output shows the starting address in memory and the size of memory allocated. For example, the starting address for the first entry is 4817f000 and the size of the memory allocated is 4096 bytes.

The shared memory window is where processes share common memory space (shared memory is faster than protected memory) but one process can write over the data of another process, causing memory corruption.

The **show processes memory *job-id*** command displays information about the text, data, and stack usage for the specified job ID.


**Note**

A process has its own private memory space. A process cannot access the memory of another process.

```
RP/0/RP0/CPU0:router# show processes memory 123
```

JID	Text	Data	Stack	Dynamic	Process
123	4096	4096	16384	241664	bgp_policy_reg_agent

The output shows the size of the text region (process executable), size of the data region (initialized and uninitialized variable), size of the process stack, and size of the dynamically allocated memory.

The **show memory compare** command displays details about heap memory usage for all processes on the router at different moments in time, comparing the results. This command is useful for detecting patterns of memory usage during events such as restarting processes, configuring interfaces, or looking for memory leaks.

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

```
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_start.out
```

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

```
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_end.out
```

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

JID	name	mem before	mem after	difference	mallocs	restart
346	top_procs	584300	587052	2752	0	
303	qsm	334144	334920	776	8	
335	sysdb_svr_local	1445844	1446004	160	4	
61	i2c_server	14464	14624	160	1	

You are now free to remove snapshot memcmp_start.out and memcmp_end.out under /p



The **show memory compare start** command takes the initial snapshot of heap memory usage for all processes on the router and sends the report to a temporary file. The **show memory compare end** command takes the second snapshot of heap memory usage for all processes on the router and sends the report to a temporary file. The snapshot taken with the **show memory compare end** command is compared with the initial snapshot when displaying the heap memory usage comparison report using the **show memory compare report** command. The **show memory compare report** command displays the heap memory comparison report. The output from the **show memory compare report** command displays details about heap memory usage for all processes on the router at different moments in time and compares the results (compares the amount of memory allocated and deallocated during a session). The report contains information about each process whose heap memory usage has changed from the time the first and second snapshots were taken. The process with the largest memory difference is listed first. The memory usage analyzer should be used on a stable system when no configuration changes are in progress.

## Resolving Process Memory Problems

The following conditions can be the cause of process memory problems:

- **Memory leak**—Occurs when a process requests or allocates memory and then forgets to free (deallocate) the memory when finished with that task. As a result, the memory block is reserved until the router is reloaded. Over time, more and more memory blocks are allocated by that process until there is no free memory available.

To detect a memory leak, use the **show memory compare end** and **show memory compare report** commands multiple times at regular intervals (either at set hours or once each day). The first **show memory compare start** command creates the process comparison table. If the difference for a specific process is constantly increasing (a process that should not be increasing), a memory leak is probable. Restart the process to free the memory and stop the memory leak using the **process restart job-id location node-id** command. If a process restart does not resolve the memory leak problem, contact Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the [Preface](#).

- **Large quantity of memory used for normal or abnormal processes**—A normal or abnormal event (for example, a large routing instability) causes the router to use an unusually large amount of processor memory for a short period of time, during which the memory has run out. The memory shortage may also be due to a combination of factors, such as:
  - A memory leak that has consumed a large amount of memory, and then a network instability pushes the free memory to zero.
  - The router does not have enough memory to begin with, but the problem is discovered only during a rare network event.

If the large memory usage is due to a normal event, install more memory. But, if the large memory usage is due to an abnormal event, fix the related problem.

- **Dead process**—A dead process is not a real process. The process is there to account for the memory allocated under the context of another process that has terminated. Restart the process if you suspect that it may be a real process, restart the process using the **process restart job-id location node-id** command. If it is not a real process or if it is a real process and it does not restart, contact Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the [Preface](#).
- **Memory fragmentation**—A process has consumed a large amount of processor memory and then released most or all of it, leaving fragments of memory still allocated either by the process or by other processes that allocated memory during the problem. If the same event occurs several times,

the memory may fragment into very small blocks, to the point where all processes requiring a larger block of memory cannot get the amount of memory that they need. This may affect router operation to the extent that you cannot connect to the router and get a prompt if the memory is badly fragmented.

If memory fragmentation is detected, shut down some interfaces. This may free the fragmented blocks. If this works, the memory is behaving normally. (You should add more memory.) If shutting down interfaces does not resolve the problem, contact Cisco Technical Support. For Cisco Technical Support contact information, see the [“Obtaining Documentation and Submitting a Service Request”](#) section on page viii in the [Preface](#).



# CHAPTER 10

## Troubleshooting Upgrading and Downgrading Software

---

This chapter describes techniques that you can use to troubleshoot the upgrading and downgrading of router software. It includes the following sections:

- [Validating and Troubleshooting ROM Monitor Software Installation, page 10-209](#)
- [Verifying and Troubleshooting the ROM Monitor Version, page 10-210](#)
- [Troubleshooting Upgrading and Downgrading ROM Monitor Software on Cisco CRS-1 Routers, page 10-213](#)
- [Troubleshooting Upgrading and Downgrading Cisco IOS XR Software Packages, page 10-214](#)

## Validating and Troubleshooting ROM Monitor Software Installation

The ROM Monitor (ROMMON) is a bootstrap program that initializes the hardware and boots the Cisco IOS XR software when you power on or reload a router. The ROMMON software is stored on the bootflash and the ROMMON variables are stored on NVRAM.

See the following sections for information on troubleshooting ROM Monitor versions and upgrading and downgrading ROM Monitor:

- [“Verifying and Troubleshooting the ROM Monitor Version” section on page 10-210](#)
- [“Troubleshooting Upgrading and Downgrading ROM Monitor Software on Cisco CRS-1 Routers” section on page 10-213](#)



### Note

A copy of the ROMMON software exists on each card. If a card fails to boot the Cisco IOS XR software, the ROMMON software takes control and places the card in ROMMON mode. Because a card in ROMMON mode is not running the Cisco IOS XR software, that card will be unavailable for normal router operations.

Use the following troubleshooting tips when installing ROMMON software:

- If any node cannot be upgraded successfully or if you see error messages similar to the following message, try reformatting the bootflash and then repeat the upgrade procedure. See *Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router* for information on the bootflash upgrade procedure.

LC/0/3/CPU0:rommon_burner[65635]: %ROMMON_BURNER-3-FILE_OP_ERR : Opening ROMMON flash

partition failed: No such file or directory in function main at line 952

- If you are upgrading only ROMMONB and the version does not change to the expected version after the upgrade, the upgrade might have failed. When the router cannot load ROMMONB, it loads ROMMONA.
- If both ROMMANB and ROMMONA are damaged due to an unexpected node reset or a power interruption during the upgrade, the affected route processors must be returned to Cisco for repair.

## Verifying and Troubleshooting the ROM Monitor Version

The ROM Monitor (ROMMON) software is known by many names such as boot software and boot image.

Although it is distributed with routers that use the Cisco IOS XR software, ROMMON is a separate program from the Cisco IOS XR software. During normal startup, the ROMMON initializes the cards, and then control passes to the Cisco IOS XR software. After the Cisco IOS XR software takes over, ROMMON is no longer in use.

A copy of the ROMMON software exists on each card. If a card fails to boot the Cisco IOS XR software, the ROMMON software takes control and places the card in ROMMON mode. Because a card in ROMMON mode is not running the Cisco IOS XR software, that card will be unavailable for normal router operations.

When the Designated Secure Domain Router System Controller (DSDRSC) in an SDR is placed in ROMMON mode, the router operations are transferred to the standby DSDRSC (if available). If both the primary and standby DSDRSCs are in ROMMON mode, then the router operations cease since the Cisco IOS XR software is no longer running.



### Caution

Ensure you have the correct ROMMON firmware on your system. See the *Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router* for the ROMMON firmware requirements and information on upgrading and downgrading ROMMON firmware and information on overriding a ROMMON boot block.

To verify and troubleshoot the ROMMON version on systems running Cisco IOS XR software, perform the following procedure.

### SUMMARY STEPS

1. **show version**
2. **show diag | inc ROM|NODE|PLIM**
3. **more nvram:/classic-rommon-var**
4. Place the designated shelf controller (DSC) in ROMMON mode:
  - a. **admin**
  - b. **config register 0x0**
  - c. **exit**
  - d. **reload**

5. **set --**
6. **environment variable(s)**  
**sync**
7. Exit ROMMON mode:
  - a. **confreg 0x102**
  - b. **reset**
8. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show version</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show version	Displays information about the router, including image names, uptime, and other system information.  Verify that the expected ROMMON version is installed. If the version is not as expected, see <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for information on upgrading or downgrading the ROM Monitor version.
Step 2	<b>show diag   inc ROM NODE PLIM</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show diag   inc ROM NODE PLIM	Displays details about the hardware and software on each node in a router.  Verify that the ROMMON version on each node is as expected. If the version is not as expected on a node, see <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for information on upgrading or downgrading the ROM Monitor version.
Step 3	<b>more nvram:/classic-rommon-var</b>  <b>Example:</b> RP/0/RP0/CPU0:router# more nvram:/classic-rommon-var	Displays the configured environment variables.  Ensure that the environment variable setting are as expected. If the environment variables are not as expected, proceed to <a href="#">Step 4</a> .
Step 4.	<b>admin</b> <b>config-register 0x0</b> <b>exit</b> <b>reload</b>  <b>Example:</b> RP/0/RP0/CPU0:router# admin RP/0/RP0/CPU0:router(admin)# config-register 0x0 RP/0/RP0/CPU0:router(admin)# exit RP/0/RP0/CPU0:router# reload	Places the DSC in ROMMON.
Step 5	<b>set</b>  <b>Example:</b> rommon B1 > set	Displays the configured environment variables.

	Command or Action	Purpose
Step 6	<b>environment variable(s)</b> <b>sync</b>  <b>Example:</b> rommon B2> IP_ADDRESS=1.1.1.1 rommon B3> IP_SUBNET_MASK=255.255.254.0 rommon B4> DEFAULT_GATEWAY=1.1.0.1 rommon B5> sync	Allows you to specify the environment variable settings.  Environment variable settings are entered in capital letters, followed by a definition. To save the current environment variable settings, enter the <b>sync</b> command. See <i>Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router</i> for more information on changing the environment variable settings.
Step 7	<b>confreg 0x102</b> <b>reset</b>  <b>Example:</b> rommon B6 > confreg 0x102 rommon B7 > reset	Exits ROMMON mode, resetting and initializing the router.
Step 8	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the <a href="#">“Obtaining Documentation and Submitting a Service Request”</a> section on page viii in the Preface.  Read the information in the <a href="#">“Troubleshooting Upgrading and Downgrading ROM Monitor Software on Cisco CRS-1 Routers”</a> section on page 10-213 before you begin any upgrade procedures.

## Examples

The following example shows that the ROMMON version is 1.38:

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

Cisco IOS XR Software, Version 3.3.0
Copyright (c) 2006 by cisco Systems, Inc.

ROM: System Bootstrap, Version 1.38(20060207:032757) [CRS-1 ROMMON],

router uptime is 11 hours, 47 minutes
System image file is "disk0:hfr-os-mbi-3.3.0/mbihfr-rp.vm"

cisco router (7457) processor with 4194304K bytes of memory.
7457 processor at 1197Mhz, Revision 1.2
```

The following example shows that the ROMMON version running on all nodes is:

```
RP/0/RP0/CPU0:router# show diag | inc ROMMON

ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
ROMMON: Version 1.500(20090929:184527) [CRS-1 ROMMON]
```

The following example shows the current environment variable settings:

```
RP/0/RP0/CPU0:router# more nvram:/classic-rommon-var

PS1 = rommon ! > , IOX_ADMIN_CONFIG_FILE = , IP_ADDRESS = 192.39.52.71, IP_SUB
```

```
NET_MASK = 255.255.255.0, DEFAULT_GATEWAY = 192.39.52.1, ? = 0, TURBOBOOT = , EA
SYBAKE = 0x0, ReloadReason = 66, BSI = 0, BOOT = disk0:hfr-os-mbi-3.3.0.1I/mbihf
r-rp.vm,1;, confreg = 0x102^@
```

The following example shows how to place the DSC in ROMMON mode:

```
RP/0/RP0/CPU0:router# admin
RP/0/RP0/CPU0:router(admin)# config-register 0x0
```

```
Successfully set config-register to 0x0 on node 0/RP0/CPU0
Successfully set config-register to 0x0 on node 0/RP1/CPU0
```

```
RP/0/RP0/CPU0:router(admin)# exit
RP/0/RP0/CPU0:router# reload
```

```
Proceed with reload? [confirm]
System Bootstrap, Version 12.0(20040624:164256) [assafb-misc1 1.14dev(0.91)] DEV
ELOPMENT SOFTWARE
Copyright (c) 1994-2004 by cisco Systems, Inc.
DRAM DIMM Slot 1: 512M found, Slot 2: Empty
MPC7450 platform with 524288 Kbytes of main memory
rommon B1 >
```

The following example show how to display the current environment variable settings:

```
rommon B1 > set
PSI=rommon ! >
TFTP_VERBOSE=2
IP_ADDRESS=1.1.1.1
IP_SUBNET_MASK=255.255.0.0
TFTP_SERVER=
DEFAULT_GATEWAY=12.25.0.1
TFTP_FILE=
CONFIG_FILE=
BOOT=disk0:hfr-os-mbi-1.0.0/mbihfr-rp.vm,1;
```

The following example shows how to set and save the environment variables:

```
rommon B2> IP_ADDRESS=10.1.1.1
rommon B3> IP_SUBNET_MASK=255.255.254.0
rommon B4> DEFAULT_GATEWAY=10.1.0.1
rommon B5> sync
```

The following example shows how to exit ROMMON mode, reset and initialize the router:

```
rommon B6> confreg 0x102
rommon B7> reset
```

## Troubleshooting Upgrading and Downgrading ROM Monitor Software on Cisco CRS-1 Routers

If you need to upgrade or downgrade the ROM Monitor (ROMMON) firmware, first upgrade the software including the FPD PIE, and then upgrade the firmware.



### Note

FPD = Field programmable device. PIE = Package installation envelope.

If you need to upgrade or downgrade the ROMMON firmware on your system, be aware of the following requirement: During the upgrade/downgrade process, you *must* be sure that each upgrade step completes successfully before continuing to the next step. (The system will provide a message when each step completes successfully.) If a step does not complete successfully and you continue to the next step, the firmware load will fail. In that case, you should attempt to install the firmware again. If the second attempt fails, you might need to return the card to the factory for repairs.

**Caution**

Pay close attention to the completion of each ROMMON upgrade/downgrade step. If you continue after any step is unsuccessful, the upgrade/downgrade will fail and the card might need to be returned to the factory to complete the firmware load.

See *Cisco IOS XR ROM Monitor Guide for the Cisco CRS-1 Router* for detailed information and procedures on upgrading and downgrading ROM Monitor software on the Cisco CRS-1.

## Troubleshooting Upgrading and Downgrading Cisco IOS XR Software Packages

See *Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router* for detailed information and procedures on upgrading and downgrading Cisco IOS XR software.





# CHAPTER 11

## Troubleshooting the Statistics Infrastructure

These sections describe the methods that you can use to diagnose problems with statistics such as wrong and missing counters:

- [Debugging Statistics Infrastructure, page 11-215](#)
- [Trace Commands for the Statistics Infrastructure, page 11-217](#)
- [Show Commands for the Statistics Infrastructure, page 11-218](#)
- [Diagnosing Problems with Statistics Values, page 11-223](#)



**Note**

Most of the commands used in the chapter requires a Cisco-support task id to execute them.

## Debugging Statistics Infrastructure

The following sections list the debug commands that are used to diagnose problems with the statistics infrastructure:

- [debug statsd api Commands, page 11-215](#)
- [debug statsd manager Commands, page 11-216](#)
- [debug statsd server errors Command, page 11-216](#)

### debug statsd api Commands

[Table 11-1](#) lists the **debug statsd api** commands that are printed from the statistics collector library and are filtered by location and process. The statistics collector library is a process which supplies the statistics to the statistics infrastructure.

**Table 11-1** *debug statsd api Commands*

Command	Description
<b>debug statsd api errors</b>	Prints any unexpected events that occurred in the statistics infrastructure. The command is safely enabled for all processes without the risk of flooding the console.
<b>debug statsd api data</b>	Prints the data that the process is sending to the statistics infrastructure. If the correct data is not being reported by the <b>show</b> commands, this command is useful to determine where the problems reside. In particular, the problems are narrowed down to a particular statistics collector by showing it is returning incorrect data. When this command is enabled, it can generate a large volume of data and hence must be restricted to a single process.

## debug statsd manager Commands

Table 11-2 lists the **debug statsd manager** commands that are printed from the statistics manager process.

**Table 11-2** *debug statsd manager Commands*

Command	Description
<b>debug statsd manager errors</b>	Prints any unexpected events that occurred in the statistics manager.
<b>debug statsd manager sysdb-edm</b>	Prints the details about the embedded device manager (EDM) in the statistics manager. The EDM is an internal service that handles requests for statistics data and returns the results. If there is a problem with retrieving statistics data through a <b>show</b> command, the details are useful. The command shows whether the request as was received by the statistics manager and whether it was interpreted correctly.
<b>debug statsd manager datarate</b>	Prints the counters received by the statistics manager and the resulting packet and byte rates that are calculated. This can be used to determine whether collectors are sending the right data and analyse unexpected rate values. It is recommended to restrict this command to a single interface.

## debug statsd server errors Command

The **debug statsd server errors** command prints any unexpected events that occurred in the statistics server.

# Trace Commands for the Statistics Infrastructure

Trace logs are buffers that wrap data which contains a history of recent events. Unlike debugging, which must be enabled, traces are always written to the logs. Therefore, the trace logs are always available after any problem has occurred. The statistics manager trace logs must always be recovered for any problem that involves the statistics infrastructure.

These sections list the **trace** commands that are used for the statistics infrastructure:

- [show statsd manager trace Command, page 11-217](#)
- [show statsd server trace Command, page 11-218](#)
- [show stats lib trace, page 11-218](#)



## Note

All the statistics trace commands are included in **show tech-support pfi** command and need not be run separately if the tech-support output has already been collected.

## show statsd manager trace Command

[Table 11-3](#) lists the trace buffers that are included in the output of the **show statsd manager trace** command. Each buffer can be displayed individually or in any combination. If no argument is specified, all buffers are displayed.

**Table 11-3** Buffer Descriptions for the **show statsd manager trace** Command

Buffer	Description
error	Lists all unexpected events that have occurred in the statistics manager. Events are stored based on uniqueness so that at least one copy of each different error message is retained even if the main buffer was wrapped.
request	Lists all the requests that have been made and the number of responses that have been received with the node and PID that the response came from. The output of the request buffer is used with the <b>show statsd collectors</b> command with the <b>brief</b> keyword, to match the response to a particular process.
collection	Lists all the updates that are received from the collector. The buffer wraps frequently because messages are sent periodically and contains only data about fairly recent events.
store	Lists the records that are added and removed from the internal table used by the statistics manager to store its data in the memory such that it can be recovered after a process restart or failover.
datatree	Lists the entries that have been added into or removed from the datatree that indexes the checkpoint table.
types	Lists the mappings that have been made between the feature and item strings and statistics type IDs.

## show statsd server trace Command

Table 11-4 lists the trace buffers that are included in the output from the **show statsd server trace** command. Each buffer can be displayed individually or in any combination. If no argument is specified, all buffers are displayed for the statistics server on the local route processor (RP). To display information for a particular location or all locations, use the **location** keyword.

**Table 11-4** Buffer Descriptions for the **show statsd server trace** Command

Buffer	Description
error	Lists all unexpected events that have occurred in the statistics server. Events are stored based on uniqueness so that at least one copy of each different error message is retained even if the main buffer was wrapped.
collectors	Lists the registrations that have been made to convey the statistics server.
contributors	Lists the registrations that have been made for the statistics contribution.

## show stats lib trace

This command can be used to see the events taking place in the Statistics library for collectors.

Table 11-5 lists the buffers that are included in the output of **show stats lib trace** command. These can be displayed individually or in any combination if required. If you do not specify any arguments, further it displays all buffers for all collectors on the local RP. To display information for a particular location or all locations add *location <foo|all>*.

**Table 11-5** Buffer Descriptions for the **show stats lib trace** Command

Buffer	Description
api	Lists all the API calls that have been made to the statistics library and their results.
error	Lists all unexpected events that have occurred in the statistics server. The error buffer has a unique component so that at least one copy of each different error message is retained even if the main buffer was wrapped.
gsp	Lists all GSP activity between the stats library and stats manager .

## Show Commands for the Statistics Infrastructure

The output from all these commands are to be collected when any statistics issue is reported. These sections provide information about the state of the statistics infrastructure:

- [show statsd manager info Command, page 11-219](#)
- [show statsd collectors Command, page 11-220](#)
- [show statsd registrations Command, page 11-222](#)
- [show statsd requests Command, page 11-222](#)

## show statsd manager info Command

The **show statsd manager info** command provides information about each of the modules that compose the statistics manager component. The command is used to display problems in the infrastructure and examine usage patterns.

The following sample output shows the start of the output of **show statsd manager info** command:

```
RP/0/RP0/CPU0:router# show statsd manager info
```

```
STATS MANAGER DATA
-----
```

[Table 11-6](#) describes the modules whose information appears in the **show statsd manager info** command output.

**Table 11-6**      *Module Descriptions for the show statsd manager info command*

Module	Description
Request	Sends the request to the Gateway Signalling Point (GSP) group of collectors. For every request, the number of processes that the message was sent to is counted, and a response is expected for each of them.
Collection	Receives all the messages from the statistics collectors.
EDM	Handles the system database (SysDB) requests from the management applications.
Data Store	Stores and checkpoints all data in the manager.
Type Store	Stores all the statistics types that are registered by the collectors.

The following sample output for the Request module shows the number of requests for which all responses were received, the number of requests that timed out before all responses were received, and the number of requests that failed for another reason:

```
Request Module
    6 successful requests,  0 timeouts,  0 errors
```

The following sample output for the Collection module shows the total number of messages and bytes that are received. The message count is broken down into periodic updates, positive responses to the statistics requests, negative responses to the statistics requests, and control messages from GSP.

**Note** Other message types may not reside in any of these categories such as type lookup requests. Therefore, the sum of these counters may not match the count of messages received.

```
Collection Module
    2415 messages received,  1039608 bytes received
    2369 updates,  6 responses,  38 nacks,  0 from GSP
```

The following sample output from the **show statsd manager info** command shows the EDM. The module outputs the number of GET, DATALIST, and FINDDATA requests that have been made and the total number of results that have been returned in the lists.

The output divides the requests into an internal namespace and a new-feature namespace. The requests for the new-feature namespace are subdivided with the latest data requests from the collectors, data cache requests, and snapshot requests.

The number of clear requests made through the EDM is also counted. The requests made by the **clear counters** command are not included.

```
EDM Module
  12 GETs, 0 DATALISTS of 0 elements, 0 FINDDATAS of 0 elements
  12 requests using internal namespace
  0 latest, 0 cached, 0 total requests using feature namespace
  0 clear requests
```

The following sample output from the **show statsd manager info** command shows the data store module. The first line of the output shows the number of bytes that is stored in the checkpointed memory and the number of records that are used. If a single statistics structure is larger than the maximum record size, the statistics structure is chained over multiple records. Therefore, the output also shows how many of the records represent the first record in a chain.

The second line shows the number of entries in the tree that are used to access the checkpoint records. Each entry holds up to three checkpoint records for the statistics data itself, as well as any rate calculation information and any snapshot that was taken, to ensure continuity over **clear counters** requests.

```
Data Store Module
  5824 bytes stored in 52 records (51 first records)
  23 entries in tree (23 with data, 23 with rates, 0 with snapshots)
```

The following sample output from the **show statsd manager info** command shows the type store module. The output provides a table of both the internally defined types and the those that are registered by the features.

```
Type Store Module
  Type  Feature name  Item name  Flags
  ----  -
  2      internal      generic    0x0
  5      internal      srp        0x0
  6      internal      IPV4_UNICAST  0x0
  7      internal      IPV4_MULTICAST  0x0
  8      internal      IPV6_UNICAST  0x0
  9      internal      IPV6_MULTICAST  0x0
  10     internal      MPLS       0x0
  11     internal      ARP        0x0
  12     internal      IPV6_ND    0x0
  13     internal      CDP        0x0
  14     internal      CLNS       0x0
  101    fib_stats     mpls_label  0x0
  104    fib_stats     mpls_ln70   0x0
  102    fib_stats     mpls_ln0    0x0
  103    fib_stats     mpls_glb    0x0
```

## show statsd collectors Command

The **show statsd collectors** command lists all the processes that have registered as a statistics collector, and can be filtered to a single node or a single process.

[Table 11-7](#) lists the keywords/options for the **show statsd collectors** command.

**Table 11-7**      **Output Descriptions from the show statsd collectors Command**

Output	Description
brief	Displays a a table of all the statistics collectors, the number of connections made from the process, the number of callback functions, and the number of registrations. In addition, the table shows the number of bytes that have been sent from the collector and the number of messages that were not sent to the manager.
default	Displays more detail about each statistics collector that includes a count of the total messages that have been sent and a breakdown into particular message types. In addition, the default output lists the callback handles for each registered callback function and provides the specified collection period, the number of registrations that are associated with the handle, and whether a bulk handler is registered or not.
detail	Displays detailed output that contains the same data as the default output. In addition, it includes a table of all the registrations made and the options that are associated with them.

The following sample output shows the brief output:

```
RP/0/RP0/CPU0:router# show statsd collectors brief location 0/0/CPU0
```

```
STATS COLLECTORS
Node      PID    Process      Conn    CB    Reg    Bytes    Errors
----
0x0      131157 statsd_server    1     2     3    19020     0
0x0      131160 fib_mgr         7    14     6     700     0
0x0      192617 ipv6_nd         1     1     0     180     0
0x0      192639 clns           1     1     0     180     0
0x0      192640 arp            1     1     1     7060     0
0x0      192665 nd_partner      1     1     1    14884     0
```

The following sample output shows the default output:

```
RP/0/RP0/CPU0:router# show statsd collectors location 0/0/CPU0 pid 131157
```

```
STATS COLLECTORS
-----
statsd_server node: 0x0    PID: 131157
  Total bytes sent: 23292  Total messages sent: 87  Errors: 0
    periodic updates: 83      on-demand replies: 0
      nacks:          1      other messages:    3
Connections: 1  Registered callbacks: 2
  1. 0x0802d9fc Options 0x0 (DEFAULT)
    Period 30 secs Bulk FALSE Total registrations 0
  2. 0x0802d9c8 Options 0x0 (DEFAULT)
    Period 30 secs Bulk TRUE  Total registrations 3
```

The following sample output shows the detailed output:

```
RP/0/RP0/CPU0:router# show statsd collectors detail location 0/0/CPU0 pid 131157
```

```
STATS COLLECTORS
-----
statsd_server node: 0x0    PID: 131157
  Total bytes sent: 25700  Total messages sent: 96  Errors: 0
    periodic updates: 91      on-demand replies: 0
      nacks:          1      other messages:    4
```

```

Connections: 1 Registered callbacks: 2
1. 0x0802d9fc Options 0x0 (DEFAULT)
   Period 30 secs Bulk FALSE Total registrations 0
2. 0x0802d9c8 Options 0x0 (DEFAULT)
   Period 30 secs Bulk TRUE Total registrations 3
   1. ifhandle 0x01000100 type 6 IPV4_UNICAST opts 0x1 (REG)
   2. ifhandle 0x01000100 type 8 IPV6_UNICAST opts 0x1 (REG)
   3. ifhandle 0x01000100 type 10 MPLS opts 0x1 (REG)

```

## show statsd registrations Command

The **show statsd registrations** command lists all the processes that return data for a specific registration. The statistics type is specified either by using the feature and item strings or by using the statistics type ID, which is found in the output from the **show statsd manager info** command. The output provides only the node and process ID (PID) values of the collector process. To locate the process name, you need the output from the **show statsd collectors** command with the **brief** keyword.

The following sample output is from the **show statsd registrations** command with the **ifname** keyword:

```
RP/0/RP0/CPU0:router# show statsd registrations ifname MgmtEth 0/0/CPU0/0 feature internal generic
```

```

STATS REGISTRATIONS
-----
ID type: 1 (ifhandle) ID: 0x01000100 feature: internal item: generic
  Node   PID           Size Options
  ----   --
  0x0    28696           188 0x11 (ABS|REG)

```

The following sample output is from the **show statsd registrations** command with the **ifhandle** keyword:

```
RP/0/RP0/CPU0:router# show statsd registrations ifhandle 0x01000100 type 6
```

```

STATS REGISTRATIONS
-----
ID type: 1 (ifhandle) ID: 0x01000100 stats type: 6
  Node   PID           Size Options
  ----   --
  0x0    131157           76 0x1 (REG)
  0x0    131160           24 0x41 (USE_SW|REG)

```

## show statsd requests Command

The **show statsd requests** command parses the ltrace buffers to provide details of the most recent statistics requests that have been handled by the statistics manager.

The following sample output is from the **show statsd requests** command:

```

RP/0/RP0/CPU0:router# show statsd requests

1 unique entries (64 possible, 0 filtered)
118 wrapping entries (1024 possible, 0 filtered, 118 total)
STATS REQUEST DATA
-----
Displaying data about the last 12 requests (0 failed)
      min / avg / max
Time taken (ms): 1 / 15 / 37

```



```

Responses:      1 / 1 / 3
Nacks:         5 / 6 / 17

```

ID	Start Time and Duration	Size	Responses	NACKs	First stats type, ID type, ID
1	Apr 18 02:58:16	37ms	0	3	17 2 (generic), 1 (ifhandle), 0x02000080
2	Apr 18 02:58:16	2ms	0	2	6 2 (generic), 1 (ifhandle), 0x02000000
3	Apr 18 02:58:16	1ms	0	2	6 2 (generic), 1 (ifhandle), 0x02000100
4	Apr 18 02:58:16	1ms	0	2	6 2 (generic), 1 (ifhandle), 0x02000200
5	Apr 18 02:58:16	6ms	0	1	5 2 (generic), 1 (ifhandle), 0x03000300
6	Apr 18 02:58:16	8ms	0	1	5 2 (generic), 1 (ifhandle), 0x03000600
7	Apr 18 02:58:16	7ms	0	1	5 2 (generic), 1 (ifhandle), 0x03000900
8	Apr 18 02:58:16	10ms	0	1	5 2 (generic), 1 (ifhandle), 0x03000c00
9	Apr 18 02:58:16	28ms	0	1	5 2 (generic), 1 (ifhandle), 0x05000300
10	Apr 18 02:58:18	28ms	0	1	5 2 (generic), 1 (ifhandle), 0x05000600
11	Apr 18 02:58:18	28ms	0	1	5 2 (generic), 1 (ifhandle), 0x05000900
12	Apr 18 02:58:18	34ms	0	1	5 2 (generic), 1 (ifhandle), 0x05000c00

The following sample output shows detailed output from the **show statsd requests** command:

```
RP/0/RP0/CPU0:router# show statsd requests detail
```

```

1 unique entries (64 possible, 0 filtered)
118 wrapping entries (1024 possible, 0 filtered, 118 total)
STATS REQUEST DATA

```

```

-----
Displaying data about the last 12 requests (0 failed)
      min / avg / max
Time taken (ms): 1 / 15 / 37
Responses:      1 / 1 / 3
Nacks:         5 / 6 / 17

```

ID	Start Time and Duration	Size	Responses	NACKs	First stats type, ID type, ID
8	Apr 18 02:58:16	10ms	0	1	5 2 (generic), 1 (ifhandle), 0x03000c00
NodeID      JID      PID Order Parts Bytes					
0x20	100	57434	4	0	24 NACK
0x20	102	61541	1	0	24 NACK
0x20	123	57450	3	0	24 NACK
0x20	139	57444	2	1	220 Data
0x20	171	57441	0	0	24 NACK
0x20	245	57428	5	0	24 NACK

## Diagnosing Problems with Statistics Values

The statistics infrastructure is the transport and storage mechanism for statistics data. Most of the problems that are seen in this area are not caused by the infrastructure itself but by the collectors that return the data or the underlying transport mechanisms.

These sections provide information about how to diagnose problems with statistics values:

- [Errors When Retrieving Data from the Statistics Manager EDM, page 11-224](#)
- [Timeouts and Delays When Retrieving Data, page 11-224](#)
- [Displaying Incorrect Rates for the show interfaces Command, page 11-225](#)
- [Displaying Incorrect Data for the show Commands, page 11-226](#)

## Errors When Retrieving Data from the Statistics Manager EDM

You need to determine whether requests from the **show** command or other management agents are being received and correctly interpreted by the statistics manager. [Table 11-8](#) lists the **debug** commands that must be enabled while running the **show** command.

**Table 11-8** *Debug Commands for the Statistics Manager*

Command	Description
<b>debug statsd manager errors</b>	Displays the errors for the statistics manager.
<b>debug statsd manager sysdb-edm</b>	Displays the EDM-related activity of the statistics manager.

Additionally, examining the diagnostic output from the **show statsd manager info** command before and after the request shows whether or not the request is being received by the statistics manager EDM.

## Timeouts and Delays When Retrieving Data

[Table 11-9](#) lists the **show** and **debug** commands that help you identify the client that is not responding.

**Table 11-9** *show Commands for Statistics Requests*

Command	Description
<b>show process blocked location all</b>	Displays the statistics manager or any statistics collector process that is being blocked.
<b>show process statsd manager</b>	Displays whether all threads are in receive state (the expected state).
<b>show statsd requests</b>	Displays the details of the recent requests, including the list of collectors that responded successfully.
<b>show statsd collectors brief</b>	Displays the entries for collectors on all nodes. If entries are missing for any node, this indicates a problem with the GSP group on that node.
<b>show statsd manager info</b>	Displays the request module information that shows whether any requests have failed or timed out.
<b>show statsd manager trace reverse</b>	<p>Displays the latest traces first. You should locate lines such as "Stats request 1 completed with 0/7 responses outstanding." For each request, you must also look at each response that was received, such as "Received response with 1 element(s) for request 3 from member with node ID 0x0, pid 217201." or "Received NACK with 0 element(s) for request 3 from member with node ID 0x0, pid 229528".</p> <p>If there is only one response outstanding, the problem resides with the collector that did not reply ( as it is blocked, or failed to join the group). Whereas, if there are many missing responses, the problem lies with the statistics GPS group, which is broken on one or more nodes.</p>

**Table 11-9** *show Commands for Statistics Requests (continued)*

Command	Description
<b>show gsp groups</b> [ <i>name group name</i>   <i>location node-id</i> ]	<p>Displays the number of members of the statistics group (for example, statsd_group) on all the nodes and the list of members on the specified node. All collectors are members of this group. Using the list of collectors together with the list of processes that actually responded to the request can reveal which process did not respond.</p> <p>The <b>show gsp groups</b> command displays a member of the group (for example, statsd_mgr_lwg). All collectors are writers to the group. Correlating the writers of the group with the members of the statsd_group name can indicate which process is having problems. If the group does not exist on one or more nodes, or if some nodes do not have the statistics manager listed as a member, there is a problem with the GSP group.</p>
<b>debug statsd api errors</b>	Prints errors if a collector is failing to send messages to the statistics manager. Enabling the <b>debug statsd api errors</b> command and repeating the request is often the fastest way to find out which collector is not responding.

If any process is blocked, use the **run attach_process -p PID** command to find out where the process is blocked.

## Displaying Incorrect Rates for the show interfaces Command

Perform this task if the packet or byte rates in the **show interfaces** command are not displayed correctly.

### SUMMARY STEPS

1. **show interfaces**
2. **show statsd collectors brief**  
or  
**show statsd registrations ifname** *interface name* **feature** *name item name*  
or  
**show statsd manager trace**
3. **debug statsd manager datarate** [*interface type instance*]
4. Contact Cisco Technical Support if the problem is not resolved.

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show interfaces</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces	Displays whether the number of packets and bytes sent and received are correct. If the packets and bytes are not correct, see <a href="#">“Displaying Incorrect Data for the show Commands”</a> section on page 11-226. If the packet and byte rates are correct, the problem resides with the statistics infrastructure. Go to the next step to diagnose further.
Step 2	<b>show statsd collectors brief</b> or <b>show statsd registrations ifname</b> <i>interface name</i> <b>feature</b> <i>name item name</i> or <b>show statsd manager trace</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show statsd collectors brief RP/0/RP0/CPU0:router# show statsd registrations ifname POS 0/1/0/0 feature internal generic RP/0/RP0/CPU0:router# show statsd registrations ifname POS 0/1/0/0 feature internal IPV4_UNICAST RP/0/RP0/CPU0:router# show statsd manager trace	Collects the data for the statistics manager. If the packet and byte rates are correct, the problem resides with the statistics infrastructure.
Step 3	<b>debug statsd manager datarate</b> [ <i>interface type instance</i> ]  <b>Example:</b> RP/0/RP0/CPU0:router# debug statsd manager datarate interface POS 0/1/0/0	Enables the <b>debug statsd manager datarate</b> command to collect the output. Turn this on for a couple of minutes. You can restrict it to a single interface to reduce the amount of output.
Step 4	Contact Cisco Technical Support.	Send all of the output collected above to Cisco Technical Support, stating the expected rate.

## Displaying Incorrect Data for the show Commands

In general, the statistics infrastructure does not do anything with the data that it is given other than adding and storing it. Therefore, the problems with the data values are almost always the result of errors in the collectors.

Perform this task to find out which collectors are returning incorrect data.

## SUMMARY STEPS

1. **show stats manager info**
2. **show statsd registrations ifname** *interface name* **type** *stats ID*
3. **show statsd collectors brief**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show stats manager info</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces	To carry out the further steps, you need the type ID. You can find this by looking at the type store module information from the <b>show statsd manager info</b> command. For example, IPV4_UNICAST has a value of 6.
Step 2	<b>show statsd registrations ifname</b> interface name <b>type</b> stats ID  <b>Example:</b> RP/0/RP0/CPU0:router# show statsd registrations ifname POS 0/1/0/0 type 300	Displays the list of processes that have registered to collect the statistics type for the interface. Retrieve the PID and location.
Step 3	<b>show statsd collectors brief</b>  <b>Example:</b> RP/0/RP0/CPU0:router# show statsd collectors brief	Displays the list of collectors to correlate the node and PID to a process name. This is the collector which is at fault.

**Note**

If the collector has specified the USE_L3 flag, the interface counters are being aggregated from the Layer 3 counters. Use the accounting keyword to find the counters for each individual protocol.

When the statistics collector process is found, you can display the data it is returning by using the **debug statsd api data** command with the **process** and **location** keywords.

In addition, if you are interested in data for a particular interface, you can verify what the statistics manager is receiving by using the **debug statsd manager datarate** command with the **interface** keyword.

The following example show how to identify the processes:

```
RP/0/RP0/CPU0:altadena#show statsd manager info | inc generic
      2      internal      generic      0x0
```

```
RP/0/RP0/CPU0:altadena#show statsd registrations ifname tenGigE 0/3/0/0 type 2
STATS REGISTRATIONS
-----
```

```
ID type: 1 (ifhandle) ID: 0x01380020 stats type: 2
Node  PID      Size  Options
----  ---
0x31   57416      188   0x1 (REG)
0x31   61558      188   0x9 (ALL|REG)
```

```
RP/0/RP0/CPU0:altadena#show statsd collectors brief | inc 57416
0x21      57416  hfr_pm      3      6      31      16840      0
0x31      57416  hfr_pm      3      6      61      27604      0
```

```
RP/0/RP0/CPU0:altadena#show statsd collectors brief | inc 61558
0x31      61558  plim_8p_10ge 1      1      3      22688      0
```





## CHAPTER 12

# Multiprotocol Label Switching Checklist

---

This chapter provides a checklist to troubleshoot the Multiprotocol Label Switching (MPLS) protocol.

You can collect information from the sample output on all routers for the following commands to troubleshoot the problems in MPLS:

- **show tech-support mpls rsvp**
- **show tech-support mpls traffic-eng**
- **show tech-support routing ospf**
- **show tech-support routing isis**

For more detailed information about the **show tech-support** commands, see *Cisco IOS XR Advanced System Command Reference for the Cisco CRS-1 Router*.

## Multiprotocol Label Switching Recommendations

Follow these recommendations to troubleshoot Multiprotocol Label Switching (MPLS):

- Verify the **show** commands for the MPLS control plane to ensure that there is no problem.
- Ensure that the hardware is programmed correctly for Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) and Label Distribution Protocol (LDP).
- Ensure that there are no communication problems in the MPLS infrastructure. If the control plane has some problems, determine if they are due to the following reasons:
  - Communication problems in the Resource Reservation Protocol (RSVP) infrastructure . In certain instances, the tunnel is not up though the topology is correct and the RSVP messages are lost.
  - Communication problems with the MPLS-TE infrastructure .
- Verify the sample output from the applicable **show tech-support** commands on all routers in the entire network topology. Copy the output to the applicable TFTP location.
  - For MPLS-TE issues, verify the output from the **show tech-support mpls traffic-eng**, **show tech-support mpls rsvp**, **show tech-support routing ospf**, and **show tech-support routing isis** commands.
  - For MPLS LDP issues, verify the output from the **show tech-support mpls ldp** command.
- Provide a topology diagram that indicates the applicable platform.
- Specify the symptom of the problem. For example, for an MPLS-TE tunnel, the problem is detected at the head, middle, or tail of a tunnel.

- Determine how the problem state was encountered.
- Verify if the problem can be reproduced. If it can be reproduced, list the detailed steps to reproduce the problem.
- Provide a decoded stack trace and core file for a process crash. Copy the core file to the applicable TFTP location.
- Save the sample output from the **show logging** command to a file. Copy the file to the applicable TFTP location. For more information, see *Cisco IOS XR System Monitoring Command Reference for the Cisco CRS-1 Router*.

## Troubleshooting L3VPN MPLS Traffic Loss

To troubleshoot the traffic loss of MPLS packets on L3VPN layer in Cisco CRS-1, perform the following procedure. The general flow of packets in a Cisco CRS-1 goes through the following sequence:

Incoming interface => Ingress pse (the forwarding asic or microcode) => Ingress queuing asic(xbma) => Switch fabric => Egress pse(forwarding asic or microcode) => Egress queuing asic(xbma) => Outgoing interface.

For detailed steps, see [Troubleshooting Transient Traffic Drop in Forwarding](#), page 3-88 in the [Troubleshooting Forwarding](#) chapter.





# CHAPTER 13

## Troubleshooting Load Balancing

---

This chapter explains the troubleshooting procedures for load balancing on the Cisco CRS-1 Router. This chapter has the following sections:

- [About Load Balancing with Cisco Express Forwarding](#)
- [Troubleshooting Layer 3 or Layer 4 Load Balancing](#)
- [Troubleshooting Layer 2 Load Balancing](#)
- [Configuration Examples for Troubleshooting Load Balancing](#)

### About Load Balancing with Cisco Express Forwarding

This document explains how Cisco IOS XR implements load balancing for Layer 2 (data link layer), Layer 3 (network layer) and Layer 4 (transport layer) flows across multiple parallel links when using Cisco Express Forwarding (CEF). This procedure explains load balancing on the Cisco CRS-1 Router only.

### Load Balancing Function

Load balancing is a Cisco IOS XR software feature that improves the utilization of parallel links by distributing traffic flows among them. A traffic flow consists of packets that have common Layer 3 and Layer 4 characteristics, such as the same source and destination IP addresses. The concept of a flow also extends to Layer 2 bundles and to virtual links such as MPLS traffic-engineering tunnels.

Cisco IOS XR uses a hash algorithm to identify individual flows and to distribute them across multiple parallel links. Each flow uses a specific link, but there are many flows in a production environment. Balancing traffic flows is similar to the earlier method of per-destination balancing, which grouped all packets by destination. Grouping packets by flow provides a better degree of granularity, and ultimately balancing, because the hash algorithm considers up to seven packet characteristics instead of just the destination. Cisco IOS XR software does not use the per-packet load balancing option, which rotates individual packets around the available links.

### Source Information for Load Balancing

Per-flow load balancing in Cisco IOS XR has two options: 3-tuple (the default) and 7-tuple. The 3-tuple option uses Layer 3 packet information to identify flows, and the 7-tuple option uses Layer 3 and Layer 4 information. A unique flow consists of packets that match all three of the 3-tuple values or all seven of the 7-tuple values.

With the 3-tuple option, the hash algorithm uses the following Layer 3 and platform-specific information to identify unique flows.

- Layer 3 information from the IP header
  - Source IP address
  - Destination IP address
- Platform-related information:
  - Router ID

With the 7-tuple option, the hash algorithm uses the following Layer 3, Layer 4, and platform-specific information to identify unique flows.

- Layer 3 information from the IP header:
  - Source IP address
  - Destination IP address
  - Protocol
- Layer 4 information from the TCP or UDP header
  - Source port
  - Destination port
- Platform-related information
  - Router ID
  - Ingress interface handle

For Layer 3 packets that do not contain Layer 4 information, the 7-tuple option substitutes a replacement value or ignores the field.

## Layer 2 Load Balancing

Load balancing at Layer 2 is also done by flow. Cisco IOS XR software provides load balancing for Layer 2 bundles only. Data is distributed to a link in proportion to the bandwidth of the link in relation to its bundle. For all IP traffic passing over a bundle interface, load balancing is done by the Forwarding Information Base (FIB) on the ingress and egress line cards. When a received packet is switched to a bundle interface, the FIB chooses which member link to use based on the source and destination IP address of the packets. The Adjacency Information Base (AIB) stores a list of adjacencies that the FIB uses to determine which member links are available for forwarding.

Layer 2 load balancing uses the 3-tuple algorithm, but uses only the source and destination IP addresses in the packet. A modulo operation on the hash result is subsequently performed using the number of entries in the load balancing table, which is driven by the link weighting.

For each member of bundle, the adjacency information includes a weight that balances the load among bundle members with different bandwidths. The theory behind the weight usage is:

- The weight reflects the bandwidth ratios of all members. For example, if you have one OC192 member and one OC48 member as part of a bundle, OC192 member is given a relative weight of 4 while OC48 has a weight as 1.
- If you have only one member in the bundle, it still gets a relative weight of 2 to initiate hashing.

- If all members have the same bandwidth, each is given relative weight of 1.
- If all members happen to be OC768, each member receives a relative weight of 2, which balances loads between two buffers.

There is nothing to configure for Layer 2 load balancing. Active links in the bundle are automatically eligible for load balancing.

## Terminology

The following terms are applicable to load balancing.

Term	Description
Prefix	Describes a destination IP network, such as 192.16.10.0/24. Cisco IOS XR adds a destination IP prefix to the routing table using information obtained from exchanging messages using a dynamic routing protocol or by manual configuration of static routes.
Path	Describes a valid route to reach a destination prefix. Cisco IOS XR assigns a cost to each path. A set of active paths to a destination prefix may have equal or unequal costs. Loads can be balanced across equal-cost paths.
Session	Describes a unidirectional communication flow between two IP nodes. All packets in a session use the same source and destination IP address.
Flow	<p>A network flow is a unidirectional sequence of packets that common header fields, which include the following:</p> <ul style="list-style-type: none"><li>• Source IP address</li><li>• Destination IP address</li><li>• IP protocol</li><li>• Source port (for example UDP or TCP port)</li><li>• Destination port</li><li>• Ingress interface</li><li>• IP type of service</li></ul>

## Troubleshooting Layer 3 or Layer 4 Load Balancing

This section describes how to troubleshoot load balancing for Layer 3 or Layer 4 unicast flows when Open Shortest Path First (OSPF) is the Interior Gateway Protocol (IGP).

- [Verifying the Routing Table Entries for Parallel Links](#)
- [Configuring Layer 4 Load Balancing](#)
- [Verifying the CEF Database and Measuring Flows](#)

# Verifying the Routing Table Entries for Parallel Links

Cisco Express Forwarding uses the path information in the IP routing table to balance traffic over multiple links. For this reason, verifying correct load balancing with Cisco Express Forwarding begins with confirming the contents of the IP routing table.



**Note**

Unlike configuration procedures, troubleshooting is not a deterministic process. This section provides a typical procedure for troubleshooting load balancing when OSPF is the IGP. For an example of this procedure, see [Troubleshooting Layer 3 or Layer 4 Load Balancing Example, page 13-240](#).

## SUMMARY STEPS

- 1. `show route destination-address`
- 2. `configure`
- 3. `router ospf process`
- 4. `maximum paths number`
- 5. `end`
- 6. `show route destination-address`
- 7. `show ospf process interface brief`
- 8. `show running-config router ospf process`
- 9. `configure`
- 10. `interface type interface-path-id`
- 11. `cost value`
- 12. `end`
- 13. `show route destination-address`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>show route destination-address</code>  <b>Example:</b> RP/0/RP0/CPU0:Router# show route 10.1.2.1	Displays the routes to a destination address. Use a destination address on another host that is reachable through the parallel links.  Verify that number of routes in the routing table equals the number of parallel links. If you have fewer routes than expected, continue with this procedure.
Step 2	<code>configure</code>  <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 3	<code>router ospf process</code>  <b>Example:</b> RP/0/RP0/CPU0:router# router ospf 200	Enters configuration mode for the OSPF process.

	Command or Action	Purpose
Step 4	<b>maximum paths</b> <i>number</i>  <b>Example:</b> RP/0/RP0/CPU0:router# maximum paths 3	Configures the maximum number of paths over which to load balance. By default, OSPF balances up to 4 equal-cost paths.
Step 5	<b>end</b>  <b>Example:</b> RP/0/RP0/CPU0:router# end	Ends the configuration process. Enter <b>yes</b> at the prompt to commit the changes.
Step 6	<b>show route</b> <i>destination-address</i>  <b>Example:</b> RP/0/RP0/CPU0:Router# show route 10.1.2.1	Displays the routes to a destination address.  Verify that number of routes in the routing table equals the number of parallel links. If you have fewer routes than expected, continue with this procedure.
Step 7	<b>show ospf process interface brief</b>  <b>Example:</b> RP/0/RP0/CPU0:Router# show ospf 200 interface brief	Shows interface information for all routes to the destination address, which displays the cost metric. OSPF balances loads over equal-cost routes only, so verify that the interfaces have equal costs. To load balance over unequal paths, use Enhanced Interior Gateway Routing Protocol or Interior Gateway Routing Protocol (EIGRP/IGRP) as the IGP instead.
Step 8	<b>show running-config router ospf process</b>  <b>Example:</b> RP/0/RP0/CPU0:Router# show running-configuration router ospf process	Displays the running configuration for the OSPF process. This is another way to determine if the interfaces have different costs.
Step 9	<b>configure</b>  <b>Example:</b> RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 10	<b>interface type interface</b>  <b>Example:</b> RP/0/RP0/CPU0:router# interface gi0/6/5/7	Enters interface configuration mode.
Step 11	<b>cost value</b>  <b>Example:</b> RP/0/RP0/CPU0:router# cost 1	Sets the cost of this interface to the same value as the cost of the other parallel interfaces. If necessary, repeat the previous step and this step for other interfaces.
Step 12	<b>end</b>  <b>Example:</b> RP/0/RP0/CPU0:router# end	Ends the configuration process. Enter <b>yes</b> at the prompt to commit the changes.
Step 13	<b>show route destination-address</b>  <b>Example:</b> RP/0/RP0/CPU0:Router# show route 10.1.2.1	Displays the routes to a destination address.  Verify that number of routes in the routing table equals the number of parallel links.

# Configuring Layer 4 Load Balancing

By default, the load balancing algorithm uses Layer 3 information only. This section describes how to configure load balancing to use both Layer 3 and Layer 4 information. Use the setting that is appropriate for your own system. Layer 4 distributes loads more evenly over the interfaces because it considers more packet parameters when defining unique flows.

## SUMMARY STEPS

- configure**
- cef load-balancing fields L4**
- end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>	Enters global configuration mode.
	<b>Example:</b> RP/0/RP0/CPU0:router# configure	
Step 2	RP/0/RP0/CPU0:Router(config)# <b>cef load-balancing fields [L3  L4]</b>	Configures the fields of the IP header that the hash algorithm uses when balancing flows across parallel links.
	<b>Example:</b> RP/0/RP0/CPU0:router(config)# cef load-balancing fields L4	
Step 3	<b>end</b>	Ends the configuration process. Enter <b>yes</b> at the prompt to commit the changes.

# Verifying the CEF Database and Measuring Flows

Cisco Express Forwarding uses the path information in the Cisco Express Forwarding (CEF) database to balance traffic over multiple links. For this reason, confirming proper CEF load balancing begins with confirming the contents of the CEF database.

## SUMMARY STEPS

- show cef ipv4** [*prefix [mask]*] | *interface-type interface-path-id*] [**detail**] [**location node-id**]
- show cef** [**ipv4** | **ipv6**] **exact-route** *source-address destination-address* [**protocol type**] [**source-port source-port**] [**destination-port destination-port**] [**ingress-interface type interface-path-id**] [**policy-class value**] [**detail** | **location node-id**]
- show interfaces** [*type interface-path-id* | **all** | **local** | **location node-id**] [**accounting** | **brief** | **detail** | **summary**]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show cef ipv4</b> [ <i>prefix [mask]</i> ]   <i>interface-type interface-path-id</i> ] [ <b>detail</b> ] [ <b>location node-id</b> ]  <b>Example:</b> RP/0/RP0/CPU0:router# show cef ipv4 10.1.2.1 detail	Displays the CEF forwarding table. Verify that it contains the same interfaces that the routing table has for this destination.
Step 2	RP/0/RP0/CPU0:Router# <b>show cef</b> [ <b>ipv4</b>   <b>ipv6</b> ] <b>exact-route</b> <i>source-address destination address</i> [ <b>protocol type</b> ] [ <b>source-port source-port</b> ] [ <b>destination-port destination-port</b> ] [ <b>ingress-interface type interface-path-id</b> ] [ <i>policy-class value</i> ] [ <b>detail</b>   <b>location node-id</b> ]  <b>Example:</b> RP/0/RP0/CPU0:router# show cef exact-route 192.168.254.1 10.1.2.1 protocol tcp source-port 5500 destination-port 80 ingress-interface gi0/6/5/4	Displays the exact route that a specific flow would take. Use this command for several flows to verify that they are distributed equally over the parallel interfaces.
Step 3	<b>show interfaces</b> [ <i>type interface-path-id</i>   <b>all</b>   <b>local</b>   <b>location node-id</b> ] [ <b>accounting</b>   <b>brief</b>   <b>detail</b>   <b>summary</b> ]  <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces accounting rates	Displays the traffic rates by interface. Use this command to verify that the simulated traffic takes the expected egress interface.

## Troubleshooting Layer 2 Load Balancing

This section describes how to troubleshoot load balancing at Layer 2. This procedure is specific to Layer 2 bundles.

- [Verifying the Bundle Status, IGP Route, and CEF Database](#)
- [Configuring Layer 4 Load Balancing](#)
- [Viewing the Expected Paths and Measuring the Flows](#)

### Verifying the Bundle Status, IGP Route, and CEF Database

Cisco Express Forwarding uses the path information in the IP routing table to balance traffic over multiple links. For this reason, confirming proper Cisco Express Forwarding load balancing begins with confirming the contents of the IP routing table. When troubleshooting a bundle, verify that the bundle is up and that the IGP route to the desired destination includes the bundle interface.



Note

Unlike configuration procedures, troubleshooting is not a deterministic process. This section provides a typical procedure for troubleshooting load balancing within an Ethernet bundle. For an example of this procedure, see [Troubleshooting Layer 2 Load Balancing Example, page 13-246](#).

SUMMARY STEPS

- 1. `show bundle {Bundle-Ether | Bundle-POS} interface-path-id`
- 2. `show route destination-address`
- 3. `show cef ipv4 prefix`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>show bundle {Bundle-Ether   Bundle-POS} interface-path-id</code>  <b>Example:</b> RP/0/RP0/CPU0:Router# show bundle bundle-ether 12	Displays the bundle status.  Verify that the bundle has the expected number of links. If not, troubleshoot the bundle first.
Step 2	<code>show route destination-address</code>  <b>Example:</b> RP/0/RP0/CPU0:Router# show route 10.1.2.1	Displays the routes to a destination address. Use a destination address on another host that is reachable through the bundle.  Verify that the route to the destination address includes the bundle interface. If not, make sure that the bundle interface is included in the IGP process configuration.
Step 3	<code>show cef ipv4 prefix</code>  <b>Example:</b> RP/0/RP0/CPU0:router# show cef ipv4 10.1.2.1	Displays the CEF forwarding table. Verify that it contains the same bundle interface that the routing table has for this subnet prefix.

Configuring Layer 4 Load Balancing

Layer 2 load balancing is achieved by inspecting the Layer 3 and Layer 4 information in the encapsulated packet. By default, the load balancing algorithm uses Layer 3 information only. This section describes how to configure load balancing to use both Layer 3 and Layer 4 information. If your system only uses Layer 3 load balancing, retain the default load balancing setting.

For the configuration procedure, see [Configuring Layer 4 Load Balancing, page 13-236](#).

Viewing the Expected Paths and Measuring the Flows

Cisco IOS XR provides a bundle utility that predicts how Layer 2 loads are balanced across member links. This is an interactive tool prompts for the information that the load balancing algorithm uses to allocate flows to member links.



## SUMMARY STEPS

1. **bundle-hash** { **Bundle-Ether** | **Bundle-Pos** } *interface-path-id*
2. **show interfaces** [*type interface-path-id* | **all** | **local** | **location node-id**] [**accounting** | **brief** | **detail** | **summary**]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>bundle-hash</b> { <b>Bundle-Ether</b>   <b>Bundle-Pos</b> } <i>interface-path-id</i>  <b>Example:</b> RP/0/RP0/CPU0:router# bundle-hash bundle-ether 12	Launches the bundle-hash utility. This is an interactive utility that prompts for the necessary information.
Step 2	<b>show interfaces</b> [ <i>type interface-path-id</i>   <b>all</b>   <b>local</b>   <b>location node-id</b> ] [ <b>accounting</b>   <b>brief</b>   <b>detail</b>   <b>summary</b> ]  <b>Example:</b> RP/0/RP0/CPU0:router# show interfaces Gi0/6/5/5	Displays interface information, which includes the traffic rates. Use this command for each link in the bundle to verify that the simulated traffic takes the expected link. Use <b>clear counters</b> to make it easier to view the traffic allocation.

# Configuration Examples for Troubleshooting Load Balancing

This section shows a practical example of troubleshooting load balancing problems, in the following sections:

- [Troubleshooting Layer 3 or Layer 4 Load Balancing Example](#)
- [Troubleshooting Layer 2 Load Balancing Example](#)

# Troubleshooting Layer 3 or Layer 4 Load Balancing Example

Table 13-1 shows the sample configuration that this section uses to describe the troubleshooting process. Two routers, Router A and Router B, connect back to back over three serial Gigabit Ethernet interfaces.

Table 13-1 Sample Configuration

Router A	Router B
<pre>interface GigabitEthernet0/6/5/5   ipv4 address 10.12.48.1 255.255.255.0 ! interface GigabitEthernet0/6/5/6   ipv4 address 10.12.44.1 255.255.255.0 ! interface GigabitEthernet0/6/5/7   ipv4 address 10.12.40.1 255.255.255.0  router ospf 200 ! area 0 !   interface GigabitEthernet0/6/5/5   interface GigabitEthernet0/6/5/6   interface GigabitEthernet0/6/5/7     cost 10</pre>	<pre>interface GigabitEthernet0/6/5/5   ipv4 address 10.12.48.2 255.255.255.0 ! interface GigabitEthernet0/6/5/6   ipv4 address 10.12.44.2 255.255.255.0 ! interface GigabitEthernet0/6/5/7   ipv4 address 10.12.40.2 255.255.255.0  router ospf 200 <b>maximum-paths 1</b> area 0 !   interface GigabitEthernet0/6/5/5   interface GigabitEthernet0/6/5/6   interface GigabitEthernet0/6/5/7     cost 10</pre>

The bold text identifies those configurations that affect load balancing.

## Verifying Parallel Links in the Routing Table

Troubleshooting of load balancing starts with the routing table, as described in the following sections:

- [Checking the IGP Routing Table](#)
- [Checking the Maximum Paths](#)
- [Checking the Route Metric](#)

### Checking the IGP Routing Table

This section shows how Router B selects one or more paths to reach Router A's loopback interface, 10.1.2.1.

By default, Open Shortest Path First (OSPF) supports four equal-cost paths to a destination. In this scenario, Router B is configured with maximum-paths equal to one, so it chooses only one path from the possible equal-cost paths. Use the **show route** command to view the current path to Router A:

```
RP/0/RP0/CPU0:RouterB# show route 10.1.2.1

Fri Sep 19 11:02:00.732 PST DST

Routing entry for 10.1.2.1/32
  Known via "ospf 200", distance 110, metric 2, type intra area
  Installed Sep 19 06:49:40.555 for 04:12:20
  Routing Descriptor Blocks
    10.12.48.1, from 10.1.2.1, via GigabitEthernet0/6/5/5
      Route metric is 2
  No advertising protos.
```

This shows that Router B selected GigabitEthernet0/6/5/5 as the single path to 10.1.2.1. Use the **show cef** command to view the corresponding CEF path:

```
RP/0/RP0/CPU0:RouterB# show cef ipv4 10.1.2.1

Fri Sep 19 11:00:21.846 PST DST

10.1.2.1/32, version 0, internal 0x40040001 (0xa9482f74) [1], 0x0 (0xa902a9cc), 0x4400
(0xa94f3300)
Updated Sep 19 06:49:40.562
remote adjacency to GigabitEthernet0/6/5/5
Prefix Len 32, traffic index 0, precedence routine (0)
  via 10.12.48.1, GigabitEthernet0/6/5/5, 4 dependencies, weight 0, class 0
    next hop 10.12.48.1
    remote adjacency
      local label 16071      labels imposed {None}
```

This verifies that CEF also uses GigabitEthernet0/6/5/5 as the egress interface for traffic forwarded to 10.1.2.1. No load balancing can occur in this configuration because there are three parallel links, but only one is allowed.

### Checking the Maximum Paths

Use the **maximum-paths** sub-command in the OSPF configuration mode to allow up to three paths in the routing table:

```
RP/0/RP0/CPU0:RouterB# configure
Fri Sep 19 11:33:27.451 PST DST
RP/0/RP0/CPU0:P2_CR-8(config)# router ospf 200
RP/0/RP0/CPU0:P2_CR-8(config-ospf)# maximum paths 3
RP/0/RP0/CPU0:P2_CR-8(config-ospf)# end
```

Use the **show route** command to confirm that the routing table contains the desired number of paths:

```
RP/0/RP0/CPU0:RouterB# show route 10.1.2.1

Fri Sep 19 11:36:23.345 PST DST

Routing entry for 10.1.2.1/32
  Known via "ospf 200", distance 110, metric 2, type intra area
  Installed Sep 19 11:34:04.284 for 00:02:19
  Routing Descriptor Blocks
    10.12.48.1, from 10.1.2.1, via GigabitEthernet0/6/5/5
      Route metric is 2
    10.12.44.1, from 10.1.2.1, via GigabitEthernet0/6/5/6
      Route metric is 2
  No advertising protos.
```

OSPF allows equal cost load balancing only, so one of the three paths must have a higher cost. To balance loads among links with unequal cost metrics, use Enhanced Interior Gateway Routing Protocol or Interior Gateway Routing Protocol (EIGRP/IGRP) as your Interior Gateway Protocol (IGP).

## Checking the Route Metric

Although OSPF is configured to support three equal cost paths, only two active paths are displayed in the routing table. Use the **show ospf interface** command to determine the reason:

```
RP/0/RP0/CPU0:RouterB# show ospf 200 interface brief
```

```
Fri Sep 19 11:39:35.212 PST DST
```

```
* indicates MADJ interface
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/6/5/5	200	0	10.12.48.2/24	1	BDR	1/1	
Gi0/6/5/6	200	0	10.12.44.2/24	1	BDR	1/1	
Gi0/6/5/7	200	0	10.12.40.2/24	<b>10</b>	BDR	1/1	

Gi0/6/5/7 has a higher cost than Gi0/6/5/6 and Gi0/6/5/5, and is therefore excluded. Use the **show run** command to confirm that Gi0/6/7 is configured with the **cost 10** command:

```
RP/0/RP0/CPU0:RouterB# show run router ospf 200
```

```
Fri Sep 19 11:46:36.142 PST DST
```

```
router ospf 200
maximum paths 3
area 0
 interface Loopback1
   passive enable
!
...
interface GigabitEthernet0/6/5/7
  cost 10
```

Use the **no cost 10** interface subcommand in the router ospf area configuration mode to restore the default cost to GigabitEthernet0/6/5/7. The **show route** command now displays three paths to the 194.168.20.0 network:

```
RP/0/RP0/CPU0:RouterB# show route 10.1.2.1
```

```
Fri Sep 19 11:52:46.942 PST DST
```

```
Routing entry for 10.1.2.1/32
  Known via "ospf 200", distance 110, metric 2, type intra area
  Installed Sep 19 11:52:34.487 for 00:00:12
  Routing Descriptor Blocks
    10.12.48.1, from 10.1.2.1, via GigabitEthernet0/6/5/5
      Route metric is 2
    10.12.44.1, from 10.1.2.1, via GigabitEthernet0/6/5/6
      Route metric is 2
    10.12.40.1, from 10.1.2.1, via GigabitEthernet0/6/5/7
      Route metric is 2
  No advertising protos.
```

The next step is to look at how Cisco Express Forwarding uses the information in the routing table to forward packets.

## Verifying IPv4/IPv6 Unicast Load Balancing

CEF load balancing for Layer 3 flows (3-tuple algorithm) is enabled by default, but Layer 4 flows are used in this example. Using Layer 4 information has the advantage of distributing flows more evenly over the parallel links, because it utilizes up to seven packet and platform parameters when calculating the egress interface. The same principles apply to Layer 3 flows, except that the load balancing algorithm checks only the source address, destination address, and router ID.

The troubleshooting process for Layer 3 and Layer 4 load balancing is described in the following sections:

- [Configuring Layer 4 Load Balancing](#)
- [Checking the CEF Database](#)
- [Verifying Load Balancing](#)

### Configuring Layer 4 Load Balancing

The default load balancing algorithm for Cisco IOS XR is Layer 3, so first enable Layer 4 (7-tuple algorithm):

```
RP/0/RP0/CPU0:P2_CR5-8# configure
RP/0/RP0/CPU0:P2_CR5-8(config)# cef load-balancing fields L4
RP/0/RP0/CPU0:P2_CR5-8 (config)# end
```

### Checking the CEF Database

Verify that the CEF database has the expected interfaces to the test destination, 10.1.2.1:

```
RP/0/RP0/CPU0:P2_CR5-8# show cef ipv4 10.1.2.1 detail

Sun Sep 21 09:41:47.457 PST DST
10.1.2.1/32, version 0, internal 0x40040001 (0xa9482f74) [1], 0x0 (0xa902a9cc),
Updated Sep 19 11:52:34.493
remote adjacency to GigabitEthernet0/6/5/5
Prefix Len 32, traffic index 0, precedence routine (0)
gateway array (0xa8e9e1b4) reference count 2, flags 0x400d00, source lsd (2),
[0 type 4 flags 0x4101000 (0xa9524590) ext 0x0 (0x0)]
LW-LDI[type=1, refc=1, ptr=0xa902a9cc, sh-ldi=0xa9524590]
via 10.12.48.1, GigabitEthernet0/6/5/5, 4 dependencies, weight 0, class 0
next hop 10.12.48.1
remote adjacency
local label 16071 labels imposed {None}
via 10.12.44.1, GigabitEthernet0/6/5/6, 4 dependencies, weight 0, class 0
next hop 10.12.44.1
remote adjacency
local label 16071 labels imposed {None}
via 10.12.40.1, GigabitEthernet0/6/5/7, 4 dependencies, weight 0, class 0
next hop 10.12.40.1
remote adjacency
local label 16071 labels imposed {None}

Load distribution: 0 1 2 (refcount 0)

Hash OK Interface Address
0 Y GigabitEthernet0/6/5/5 remote
1 Y GigabitEthernet0/6/5/6 remote
2 Y GigabitEthernet0/6/5/7 remote
```

This shows that the CEF database does contain the three parallel links to 10.1.2.1 and shows the associated hash bucket for each one. Therefore, CEF can correctly balance loads across these three interfaces.

## Verifying Load Balancing

Now you can verify that the flows are actually distributed among the parallel interfaces. The **show cef exact-route** command displays the expected result of the hash calculation. If you are using the 3-tuple configuration, you only need the required parameters. The protocol, source-port, destination-port, and ingress-interface only apply to 7-tuple configurations.

In this example, a traffic generator is sending TCP packets to 10.1.2.1 from 192.168.254.1 through interface Gi0/6/5/4. The source TCP port is 5500 and the destination port is 80. The following command displays the calculated egress interface for this flow:

```
RP/0/RP0/CPU0:P2_CRS-8# show cef exact-route 192.168.254.1 10.1.2.1 protocol tcp
source-port 5500 destination-port 80 ingress-interface gi0/6/5/4
```

```
Tue Sep 23 03:17:36.339 PST DST
10.1.2.1/32, version 0, internal 0x40040001 (0xa9482f74) [1], 0x0 (0xa902a9cc), 0x4400
(0xa9db444c)
Updated Sep 19 11:52:34.493
remote adjacency to GigabitEthernet0/6/5/5
Prefix Len 32, traffic index 0, precedence routine (0)
via GigabitEthernet0/6/5/5
```

CEF load balancing sends this particular flow out of interface GigabitEthernet0/6/5/5. You can verify that a flow with the specified characteristics is actually using GigabitEthernet0/6/5/5 using the following command:

```
RP/0/RP0/CPU0:P2_CRS-8# show interface accounting rates
```

```
Tue Sep 23 05:57:03.780 PST DST
GigabitEthernet0/6/5/4
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_UNICAST	43000	95	0	0
IPV4_MULTICAST	0	0	0	0

```
GigabitEthernet0/6/5/5
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_UNICAST	0	0	43000	92
IPV4_MULTICAST	0	0	0	0

```
GigabitEthernet0/6/5/6
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_MULTICAST	0	0	0	0
ARP	0	0	0	0

```
GigabitEthernet0/6/5/7
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_MULTICAST	0	0	0	0

This shows that the ingress data rate on GigabitEthernet0/6/5/4 is 43000 bps and that the egress data rate on interface GigabitEthernet0/6/5/5 is also 43000 bps, as expected.

Other flows usually use a different egress interface, because the hash algorithm produces a different hash value. For example, a second flow that is identical to the first one except for the destination port is forwarded out a different interface, GigabitEthernet0/6/5/7:

```
RP/0/RP0/CPU0:P2_CRS-8# show cef exact-route 192.168.254.1 10.1.2.1 protocol tcp
source-port 5500 destination-port 23 ingress-interface gi0/6/5/4
```

```
Tue Sep 23 04:03:25.145 PST DST
10.1.2.1/32, version 0, internal 0x40040001 (0xa9482f74) [1], 0x0 (0xa902a9cc), 0x4400
(0xa9db444c)
Updated Sep 19 11:52:34.493
remote adjacency to GigabitEthernet0/6/5/7
Prefix Len 32, traffic index 0, precedence routine (0)
via GigabitEthernet0/6/5/7
```

After the traffic counters stabilize, the display now shows that the ingress flow on GigabitEthernet0/6/5/4 has doubled and that that traffic is distributed equally between two egress interfaces, GigabitEthernet 0/6/5/5 and 0/6/5/7, as expected:

```
RP/0/RP0/CPU0:P2_CRS-8# show interface accounting rates
```

```
Tue Sep 23 05:57:03.780 PST DST
GigabitEthernet0/6/5/4
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_UNICAST	<b>86000</b>	192	0	0
IPV4_MULTICAST	0	0	0	0
ARP	0	0	0	0

```
GigabitEthernet0/6/5/5
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_UNICAST	0	0	<b>43000</b>	96
IPV4_MULTICAST	0	0	0	0

```
GigabitEthernet0/6/5/6
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_MULTICAST	0	0	0	0

```
GigabitEthernet0/6/5/7
```

	Ingress		Egress	
Protocol	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_UNICAST	0	0	<b>43000</b>	92
IPV4_MULTICAST	0	0	0	0

In an actual system, there is much more live traffic, so to the extent possible, reduce existing traffic so you can more easily view the test traffic. These results were obtained with a traffic generator that creates TCP flows, but you can obtain similar results using the **ping** command. Just make sure that the ping packets pass through the router with load balancing interfaces to reach the destination IP address.

# Troubleshooting Layer 2 Load Balancing Example

Table 13-2 shows the sample configuration that this section uses to describe the troubleshooting process for Layer 2 load balancing. Two routers, Router A and Router B, connect back to back over three serial Gigabit Ethernet interfaces that are grouped into a bundle.

Table 13-2 Sample Configuration for Layer 2 Load Balancing

Router A	Router B
<pre>interface Bundle-Ether12   description Connected to Router B   ipv4 address 10.12.48.1 255.255.255.0 ! interface GigabitEthernet0/6/5/5   bundle id 12 mode active ! interface GigabitEthernet0/6/5/6   bundle id 12 mode active ! interface GigabitEthernet0/6/5/7 bundle id 12 mode active ! router ospf 200 ! area 0 ! interface GigabitEthernet0/6/5/5 interface GigabitEthernet0/6/5/6 interface GigabitEthernet0/6/5/7</pre>	<pre>interface Bundle-Ether12   description Connected to Router A   ipv4 address 10.12.48.2 255.255.255.0 ! interface GigabitEthernet0/6/5/5   bundle id 12 mode active ! interface GigabitEthernet0/6/5/6   bundle id 12 mode active ! interface GigabitEthernet0/6/5/7   bundle id 12 mode active ! router ospf 200 area 0 ! interface GigabitEthernet0/6/5/5 interface GigabitEthernet0/6/5/6 interface GigabitEthernet0/6/5/7</pre>

## Verifying the Bundle Status

The configuration in Table 13-2 shows that a bundle is configured. Enter the show bundle command to verify the configuration:

```
RP/0/RP0/CPU0:RouterB# show bundle bundle-ether12

Fri Oct 24 02:31:14.331 PST DST
State: 0 - Port is Detached. 1 - Port is Waiting.
       2 - Port is Attached. 3 - Port is Collecting.
       4 - Port is Distributing.

Bundle-Ether12
  Bandwidth (Kbps)
  Effective    Available    MAC address    Min active    Max active
  -----
          3000000          3000000    0015.6358.b902          1          1          32

  Port          State    Port ID          B/W (Kbps)    MAC address
  -----
  Gi0/6/5/5      4        0x8000, 0x0005    1000000    0015.6358.be38
  Gi0/6/5/6      4        0x8000, 0x0006    1000000    0015.6358.be39
  Gi0/6/5/7      4        0x8000, 0x0007    1000000    0015.6358.be3a
```

This verifies that the bundle is active and is using the three expected interfaces (ports). If the bundle on your system is not active, troubleshoot that problem first. It might not have the minimum number of active links or minimum bandwidth to come up.



## Checking the IGP Routing Table and CEF Database

This section shows how to verify that Router B uses Bundle-Ether12 to reach Router A's loopback interface, 10.1.2.1. Use the **show route** command to view the current path to RouterA:

```
RP/0/RP0/CPU0:RouterB# show route 10.1.2.1

Fri Oct 24 02:23:48.643 PST DST

Routing entry for 10.1.2.1/32
  Known via "ospf 200", distance 110, metric 2, type intra area
  Installed Oct 22 02:47:06.383 for 1d23h
  Routing Descriptor Blocks
    10.12.48.2, from 10.1.2.1, via Bundle-Ether12
      Route metric is 2
  No advertising protos.
```

This shows that Router B selected Bundle-Ether12 as the single path to 10.1.2.1. Use the **show cef** command to view the corresponding CEF path:

```
RP/0/RP0/CPU0:RouterB# show cef ipv4 10.1.2.1

Fri Oct 24 02:25:13.453 PST DST
10.1.2.1/32, version 0, internal 0x40040001 (0xa93159cc) [1], 0x0 (0xa909b448),
Updated Oct 22 02:47:06.391
Prefix Len 32, traffic index 0, precedence routine (0)
  via 10.12.48.2, Bundle-Ether12, 2 dependencies, weight 0, class 0
    next hop 10.12.48.2
      local adjacency
        local label 16028          labels imposed {None}
```

This verifies that CEF also uses Bundle-Ether12 as the egress interface for traffic forwarded to 10.1.2.1. No Layer 3 or Layer 4 load balancing can occur in this configuration because there is only one link. Layer 2 load balancing can occur however, because the single link is a bundle of three separate interfaces.

## Verifying Bundle Load Balancing

Cisco IOS XR software has a bundle utility that predicts how load balancing will distribute flows among the bundle links. This is an interactive utility that prompts for the information that the load balancing algorithm uses to assign flows to links. For more information, see *Cisco IOS XR Interface and Hardware Component Command Reference*.

This example shows how to use the bundle hash utility to determine the link that a specific Layer 4 flow would take, and the links that other flows in the subnet would take:

```
RP/0/RP0/CPU0:RouterB# bundle-hash bundle-ether 12

Fri Oct 24 07:03:09.163 PST DST
Specify load-balance configuration (L3/3-tuple or L4/7-tuple) (L3,L4): L4
Single SA:SP/DA:SP pair (IPv4,IPv6) or range (IPv4 only): S/R [S]: s

Enter bundle type IP V4 (1) or IP V6 (2): 1
Enter source IP V4 address: 192.168.254.1
Enter destination IP V4 address: 10.1.2.1

Ingress interface --
- physical interface format: [ POS | GigabitEthernet | TenGigE ]R/S/I/P
- bundle interface format:   [ Bundle-Ether | Bundle-POS ]bundle-id
Enter ingress interface: GigabitEthernet0/6/5/4
```

```

Enter L4 protocol (TCP,UDP,SCTP,L2TPV3,NONE): TCP
Enter src port: 5500
Enter destination port: 80
Compute destination address set for all members? [y/n]: y
Enter subnet prefix for destination address set: 10.0.0.0
Enter bundle IP V4 address [192.168.254.1]: 10.12.48.1

```

**S/D pair 192.168.254.1:5500/10.1.2.1:80 -- Link hashed to is GigabitEthernet0/6/5/5**

```

Destination address set for subnet 10.0.0.0:
S/D pair 192.168.254.1:5500/10.0.0.3:80 hashes to link GigabitEthernet0/6/5/5
S/D pair 192.168.254.1:5500/10.0.0.9:80 hashes to link GigabitEthernet0/6/5/6
S/D pair 192.168.254.1:5500/10.0.0.1:80 hashes to link GigabitEthernet0/6/5/7

```

This shows that an ingress flow from 192.168.254.2:5500 on interface GigabitEthernet0/6/5/4 to destination 10.1.2.1:80 would use egress interface GigabitEthernet0/6/5/5. You can verify that this occurs by viewing the traffic for each of the bundle links, as shown in the following excerpts from **show interfaces**:

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/6
```

```

Tue Oct 28 03:43:18.783 PST DSTGigabitEthernet0/6/5/5 is up, line protocol is up
...
16196 packets output, 875154 bytes, 0 total output drops
Output 0 broadcast packets, 12 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/6
```

```

Tue Oct 28 03:43:18.783 PST DST
GigabitEthernet0/6/5/6 is up, line protocol is up
...
3 packets output, 372 bytes, 0 total output drops
Output 0 broadcast packets, 3 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/7
```

```

Tue Oct 28 03:43:23.482 PST DST
GigabitEthernet0/6/5/7 is up, line protocol is up
...
3 packets output, 372 bytes, 0 total output drops
Output 0 broadcast packets, 3 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

This verifies that the flow is using GigabitEthernet0/6/5/5, as expected. The other two links in the bundle have incidental traffic.

Repeat this procedure for other flows. For example, for a flow exactly the same as the first except that the destination port is 23, the bundle hash utility provides the following information:

```
RP/0/RP0/CPU0:RouterB# bundle-hash bundle-ether 12
```

```

Tue Oct 28 03:56:48.786 PST DST
Specify load-balance configuration (L3/3-tuple or L4/7-tuple) (L3,L4): L4
Single SA:SP/DA:SP pair (IPv4,IPv6) or range (IPv4 only): S/R [S]: S

```

```

Enter bundle type IP V4 (1) or IP V6 (2): 1
Enter source IP V4 address: 192.168.254.1
Enter destination IP V4 address: 10.1.2.1

Ingress interface --
- physical interface format: [ POS | GigabitEthernet | TenGigE ]R/S/I/P
- bundle interface format:   [ Bundle-Ether | Bundle-POS ]bundle-id
Enter ingress interface: GigabitEthernet0/6/5/4

Enter L4 protocol (TCP,UDP,SCTP,L2TPV3,NONE): TCP
Enter src port: 5500
Enter destination port: 23
Compute destination address set for all members? [y/n]: y
Enter subnet prefix for destination address set: 10.0.0.0
Enter bundle IP V4 address [192.168.254.1]: 10.12.48.1

S/D pair 192.168.254.1:5500/10.1.2.1:23 -- Link hashed to is GigabitEthernet0/6/5/7

```

```

Destination address set for subnet 10.0.0.0:
S/D pair 192.168.254.1:5500/10.0.0.2:23 hashes to link GigabitEthernet0/6/5/5
S/D pair 192.168.254.1:5500/10.0.0.1:23 hashes to link GigabitEthernet0/6/5/6
S/D pair 192.168.254.1:5500/10.0.0.4:23 hashes to link GigabitEthernet0/6/5/7

```

This shows that an ingress flow from 192.168.254.2:5500 on interface GigabitEthernet0/6/5/4 to destination 10.1.2.1:23 would use egress interface GigabitEthernet0/6/5/7. You can verify that this occurs by viewing the traffic for each of the bundle links, as shown in the following excerpts from **show interfaces**:

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/5
```

```

Tue Oct 28 04:01:57.568 PST DST
GigabitEthernet0/6/5/5 is up, line protocol is up
...
1 packets output, 94 bytes, 0 total output drops
Output 0 broadcast packets, 1 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/6
```

```

Tue Oct 28 04:02:01.254 PST DST
GigabitEthernet0/6/5/6 is up, line protocol is up
...
0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

```
RP/0/RP0/CPU0:RouterB# show interface Gi0/6/5/7
```

```

Tue Oct 28 04:02:04.704 PST DST
GigabitEthernet0/6/5/7 is up, line protocol is up
...
18190 packets output, 982260 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

This verifies that the flow to port 23 is using GigabitEthernet0/6/5/7, as expected.

For more information on implementing Cisco Express Forwarding on Cisco IOS XR Software, see *Cisco IOS XR IP Addresses and Services Configuration Guide for the Cisco CRS-1 Router*. For more information on Cisco Express Forwarding commands on Cisco IOS XR Software, see *Cisco IOS XR IP Addresses and Services Command Reference for the Cisco CRS-1 Router*.

For more information on Cisco IOS Load Balancing Troubleshooting, see *Troubleshooting Load Balancing Over Parallel Links Using Cisco Express Forwarding*.



## INDEX

---

### B

blocked process (see process block) [8-188](#)  
booting  
    configuration register [2-73](#)

---

### C

capture software packets command [1-58](#)  
Catalyst switch  
    enable command [6-158](#)  
    show running-config command [6-158](#)  
cfs check command [1-40](#)  
chassis numbering [2-73](#)  
Cisco Technical Support [1-70](#)  
CLI access [1-2](#)  
commit confirmed command [1-38](#)  
config-register command [6-150, 6-156, 6-161, 10-211](#)  
configuration register [2-73](#)  
confreg command [2-73, 6-151, 6-156, 6-162, 10-212](#)  
console port, troubleshooting [2-74](#)  
controllers fabric link port command [4-123](#)  
control plane Ethernet network  
    overview [6-147](#)  
CPU utilization [8-185](#)

---

### D

debug shelfmgr boot command [6-149](#)  
describe hostname command [1-39](#)  
diagnostic commands [7-166](#)  
dumppcore command [8-173](#)

---

### E

environment variable command [10-212](#)

---

### F

Fabric  
    Overview [4-111](#)  
fabric  
    cleaning optical fibers [4-124](#)  
    down fabric planes [4-126](#)  
    fabric plane state [4-115](#)  
    up fabric planes [4-121](#)  
follow command [8-173, 8-188, 8-192](#)

---

### I

install verify command [1-30](#)

---

### L

logging archive command [1-60](#)

---

### M

man command [1-7](#)  
MIBs [1-60](#)  
monitor controller command [1-68](#)  
monitor interface command [1-67](#)  
monitor processes command [1-69](#)  
monitor threads command [1-70](#)  
more nvram command [10-211](#)

## N

### network

documenting [1-2](#)

no controllers fabric plane shutdown command [4-127](#)

numbering, chassis [2-73](#)

## O

optical interfaces [5-139](#)

## P

ping command [7-166](#)

port, console [2-74](#)

### process

reload [8-195](#)

timeouts [8-185](#)

process block [8-188](#)

process mandatory command [8-180](#)

process restartability (see restartability) [8-179](#)

process restart command [8-189, 8-193](#)

### process timeouts

SNMP [8-194](#)

prompt, router [1-2](#)

## R

rack numbering [2-73](#)

reload (see process reload) [8-195](#)

reset command [2-73](#)

restartability [8-179](#)

router prompt [1-2](#)

## S

show_bcm_links command [6-151, 6-156](#)

show adjacency command [3-87](#)

show adjacency detail hardware command [3-87](#)

show adjacency hardware trace location command [3-87](#)

show adjacency ipv4 nexthop command [3-86](#)

show adjacency ipv4 nexthop detail hardware command [3-87](#)

show adjacency remote detail command [3-86, 3-89](#)

show adjacency remote detail hardware command [3-86](#)

show adjacency trace client command [3-87](#)

show adjacency trace command [3-87](#)

show arp command [1-5, 3-86, 3-89](#)

show arp traffic location command [3-86, 3-89](#)

show asic-errors command [1-50](#)

show captured packets command [1-58](#)

show cef ipv4 detail command [3-83](#)

show cef ipv4 detail location command [3-84](#)

show cef ipv4 hardware egress command [3-84](#)

show cef ipv4 hardware ingress command [3-84](#)

show cef ipv4 interface command [3-85](#)

show cef ipv4 summary command [3-85](#)

show cef ipv4 trace command [3-85](#)

show cef platform trace ipv4 all command [3-85](#)

show cfgmgr trace command [1-40](#)

show config commit history command [1-37](#)

show configuration commit changes command [1-37, 1-40](#)

show configuration commit list command [1-38](#)

show configuration failed command [1-47](#)

show configuration failed startup command [1-40](#)

show configuration history commit command [1-40](#)

show context command [1-11, 1-15](#)

show controllers backplane ethernet clients all command [6-150](#)

show controllers backplane ethernet command [6-156, 6-161](#)

show controllers backplane ethernet local clients statistics command [6-150](#)

show controllers fabric bundle port all command [4-128](#)

show controllers fabric connectivity all command [4-128](#)

show controllers fabric link port command [4-117, 4-122, 4-123, 8-188, 8-192](#)

- show controllers fabric plane all detail command [4-116, 4-127](#)
  - show controllers fabric plane all statistics command [4-116](#)
  - show controllers fabric plane statistics detail command [4-117](#)
  - show controllers fabric sfe command [4-118, 4-128](#)
  - show controllers plim asic statistics command [5-132](#)
  - show controllers stats command [5-132](#)
  - show controllers switch ports command [6-161](#)
  - show controllers switch ports location command [6-155, 6-160](#)
  - show controllers switch statistics command [6-150, 6-155, 6-161](#)
  - show diag command [10-211](#)
  - show environment command [1-15](#)
  - show exception command [8-173](#)
  - show fault manager metric command [8-182](#)
  - show hw-module subslot counters mac command [5-132](#)
  - show im chains command [3-92](#)
  - show imds interface brief [3-92](#)
  - show install active command [1-32](#)
  - show install command [1-28](#)
  - show install committed command [1-32, 1-33](#)
  - show interface brief command [1-16](#)
  - show interface command [5-132, 5-144](#)
  - show interfaces brief command [1-4](#)
  - show interfaces command [1-5](#)
  - show ipv4 interface command [1-5](#)
  - show logging command [1-14, 7-165](#)
  - show memory command [9-204](#)
  - show memory compare command [9-202, 9-204](#)
  - show memory heap command [1-15](#)
  - show memory heap summary command [9-202](#)
  - show memory summary command [1-15, 9-202](#)
  - show netio idb command [3-91, 5-132](#)
  - show platform command [1-10, 1-14, 2-74, 6-149, 6-155, 6-160](#)
  - show processes aborts command [8-182](#)
  - show processes all command [8-180](#)
  - show processes blocked command [8-178, 8-181, 8-183, 8-188, 8-192](#)
  - show processes boot command [8-175](#)
  - show processes failover command [8-177](#)
  - show processes memory command [9-198, 9-203](#)
  - show processes pidin command [8-183](#)
  - show processes startup command [8-176](#)
  - show process memory command [9-204](#)
  - show route ipv4 command [3-83](#)
  - show running-config [1-39](#)
  - show running-config command [1-14, 1-15, 1-35, 4-127, 6-159](#)
  - show spanning tree command [2-80](#)
  - show spantree mst brief command [6-155](#)
  - show sysdb trace command [1-36](#)
  - show sysdb trace verification command [1-39](#)
  - show sysmgr trace verbose command [8-182](#)
  - show system verify command [1-16, 7-165](#)
  - show tbn hardware ipv4 unicast dual detail command [3-92](#)
  - show tech support command [1-2](#)
  - show tech-support command [7-165](#)
  - show uidb data command [3-91](#)
  - show uidb index command [3-91](#)
  - show users command [1-13](#)
  - show version command [1-14, 1-28, 10-211](#)
  - show watchdog command [9-200](#)
  - show watchdog memory-state command [9-198, 9-203](#)
  - SNMP [1-60](#)
    - timeouts [8-194](#)
  - snmp-server community command [1-60](#)
  - sohw hw-module subslot counters framer command [5-132](#)
  - Switch fabric (see Fabric) [4-111](#)
  - sync command [10-212](#)
- 
- ## T
- timeouts (see process timeouts) [8-185](#)
  - top command [1-10, 8-183](#)
  - top processes command [1-15](#)
  - trace commands [1-56](#)
  - traceroute commands [7-166](#)

transport input telnet command [1-4](#)

---

## W

watchdog monitor cpu-hog persistent timeout  
command [9-201](#)

watchdog threshold memory command [9-199](#)