

Troubleshoot Router Crashes

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Get Information About the Crash](#)

[Types of Crashes](#)

[Router Module Crashes](#)

[Examples of Output which Indicate the Crash](#)

[Information to Collect if You Open a TAC Service Request](#)

Introduction

This document describes how to troubleshoot router crashes. A "system crash" is a situation where the system has detected an unrecoverable error and has restarted itself. The errors that cause crashes are typically detected by processor hardware, which automatically branches to special error handling code in the ROM monitor. The ROM monitor identifies the error, prints a message, saves information about the failure, and restarts the system.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Get Information About the Crash

When the router crashes, it is extremely important to gather as much information as possible about the crash before you manually reload or power-cycle the router. All information about the crash, except that which has been successfully stored in the crashinfo file, is lost after a manual reload or power-cycle. These outputs give some indication and information on the crash.

If you have the output of a **show version**, **show stacks**, **show context**, or **show tech support** command from your Cisco device, you can use [Cisco CLI Analyzer](#) to display potential issues and fixes. In order to use [Cisco CLI Analyzer](#), you must be a [registered](#) customer, be logged in, and

have JavaScript enabled.

Command	Description
---------	-------------

show version	This command first appeared in Cisco IOS [®] Software Release 10.0. The show version EXEC command displays the configuration of the system hardware, the software version, the name of the system, and sources of configuration files and software images, the router uptime, and information on how the system has been restarted. IMPORTANT: If the router is reloaded after the crash (for example, if it has been power-cycled or the reload command has been issued), this information will be lost, so try to collect it before reloading!
---------------------	--

show stacks	This command first appeared in Cisco IOS Software Release 10.0. The show stacks EXEC command is used to monitor the stack usage of processes and interrupt routines. The show stacks output is one of the most indispensable sources of information to collect when the router crashes. IMPORTANT: If the router is reloaded after the crash (for example, through power cycle or the reload command), this information will be lost so try to collect it before reloading!
--------------------	---

show context	This command first appeared in Cisco IOS Software Release 10.3. The show context EXEC command is used to display information stored in nonvolatile RAM (NVRAM) when an exception occurs. Context information is specific to processors and architectures, whereas software version and uptime information are not. Context information for different router types could therefore differ. The output displayed from the show context command includes:
---------------------	--

- the reason for the system reboot.
- stack trace.
- software version.
- signal number, code, and router uptime information.
- all the register contents at the time of the crash.

show tech-support	This command first appeared in Cisco IOS Software Release 11.2. This command is useful for collecting general information about the router when you report a problem. It includes:
--------------------------	--

- **show version**
- **show running-config**
- **show stacks**
- **show interface**
- **show controller**
- **show process cpu**
- **show process memory**
- **show buffers**

console log	If you are connected to the console of the router at the time of the crash, you will see some output like this during the crash:
--------------------	--

```
*** System received a Software forced crash ***
signal= 0x17, code= 0x24, context= 0x619978a0
PC = 0x602e59dc, Cause = 0x4020, Status Reg = 0x34008002
DCL Masked Interrupt Register = 0x000000f7
DCL Interrupt Value Register = 0x00000010
MEMD Int 6 Status Register = 0x00000000
```

syslog	Keep this information and the logs before it. Once the router comes up again, do not forget to get the show stacks output.
---------------	---

syslog	If the router is set up to send logs to a syslog server, you will see some information on what happened before the crash on the syslog server. However, when the router is crashing, it may not be able to send the most useful information to this syslog server. So most of the time, syslog output is not very useful for troubleshooting crashes.
---------------	--

<u>crashinfo</u>	The crashinfo file is a collection of useful information related to the current crash, stored in bootflash or flash memory. When a router crashes due to data or stack corruption, more information is needed to debug this type of crash than just the output from the normal show stacks command. The crashinfo is written by default to bootflash:crashinfo on the Cisco
----------------------------------	---

12000 Gigabit Router Processor (GRP), the Cisco 7000 and 7500 Route Switch Processor (RSPs), and the Cisco 7200 series routers. For the Cisco 7500 Versatile Interface Processor (VIP2), this file is stored by default to **bootflash:vip2_slot_no_crashinfo** where the *slot_n* is the VIP2 slot number. For the Cisco 7000 Route Processor (RP), the file is stored by default to **flash:crashinfo**. For more details, see [Retrieving Information from the Crashinfo File](#).

core dump A core dump is a full copy of the router's memory image. This information is not necessary for troubleshooting most types of crashes, but it is highly recommended when filing a new bug report. You may need to enable some debugs to add more information into the core dump such as `debug sanity`, `scheduler heapcheck process`, and `memory check-interval 1`. For more details, see [Creating Core Dumps](#).

rom monitor The router might end up in ROM monitor after a crash when its config-register setting ends in 0. If the processor is a 68k, the prompt will be ">". You can get the stack trace with the **k** command. If the processor is a reduced instruction set computing (RISC), the prompt will be "rommon 1>". Get the output of **stack 50** or **show context**.

Types of Crashes

The **show version** and **show stacks** commands provide you with output that gives you an indication of the type of the crash that occurred, such as bus error, or software forced crash. You can also get crash type information from the **crashinfo** and **show context** commands. For some later Cisco IOS Software Releases, the crash reasons are not clearly indicated (for example, you see "Signal = x" where x is a number). Refer to [Versatile Interface Processor Crash Reason Codes](#) to translate this number into something meaningful. For example, "Signal = 23" translates to a software forced crash. Follow these links to troubleshoot the specific type of crash your router experiences:

- [Abort](#)
- [Address Error](#)
- [Bus Error](#)
- [Cache Error Exception](#)
- [Error - Level <x>](#)
- [Format Error](#)
- [Illegal Instruction](#)
- [Illegal Opcode Exception](#)
- [Jump to Zero Error](#)
- [Line Emulator Trap](#)
- [Power-On](#)
- [Processor Memory Parity Error](#)
- [Reserved Exception](#)
- [Restarted by Error](#)
- [Segmentation Violation Exception](#)
- [Shared Memory Parity Error](#)
- [SIGTRAP](#)
- [Software-forced Crash](#)
- [Trace Trap](#)
- [Undefined Trap](#)
- [Unexpected Hardware Interrupt](#)
- [Unknown Failure](#)
- [Unknown Reload Cause](#)

- [Watchdog Timeout](#)
- [Write Bus Error Interrupt](#)

Router Module Crashes

Sometimes, only a specific router module crashes, and not the router itself. Here are some documents that describe how to troubleshoot crashes on some router modules:

- [Troubleshooting VIP Crashes](#)
- [Troubleshooting SAR Crashes on PA-A3](#)
- [Troubleshooting Line Card Crashes on the Cisco GSR12000 Series](#)

Examples of Output which Indicate the Crash

```
Router#show version
```

```
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-PV-M), Version 12.0(10.6)ST, EARLY DEPLOYMENT
MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Fri 23-Jun-00 16:02 by richv
Image text-base: 0x60010908, data-base: 0x60D96000
```

```
ROM: System Bootstrap, Version 12.0(19990806:174725), DEVELOPMENT SOFTWARE
BOOTFLASH: RSP Software (RSP-BOOT-M), Version 12.0(9)S, EARLY DEPLOYMENT
RELEASE SOFTWARE (fc1)
```

```
Router uptime is 20 hours, 56 minutes
System returned to ROM by error - a Software forced crash, PC 0x60287EE8
System image file is "slot0:rsp-pv-mz.120-10.6.ST"
```

```
cisco RSP8 (R7000) processor with 131072K/8216K bytes of memory.
R7000 CPU at 250Mhz, Implementation 39, Rev 1.0, 256KB L2, 2048KB L3 Cache
Last reset from power-on
G.703/E1 software, Version 1.0.
G.703/JT2 software, Version 1.0.
X.25 software, Version 3.0.0.
Chassis Interface.
1 EIP controller (6 Ethernet).
1 VIP2 R5K controller (1 FastEthernet)(2 HSSI).
6 Ethernet/IEEE 802.3 interface(s)
1 FastEthernet/IEEE 802.3 interface(s)
2 HSSI network interface(s)
2043K bytes of non-volatile configuration memory.
20480K bytes of Flash PCMCIA card at slot 0 (Sector size 128K).
16384K bytes of Flash internal SIMM (Sector size 256K).
No slave installed in slot 7.
Configuration register is 0x2102
```

```
Router#show stacks
```

```
Minimum process stacks:
Free/Size   Name
5188/6000   CEF Reloader
9620/12000  Init
5296/6000   RADIUS INITCONFIG
5724/6000   MDFS Reload
2460/3000   RSP memory size check
8176/9000   DHCP Client
```

Interrupt level stacks:

Level	Called	Unused/Size	Name
1	163	8504/9000	Network Interrupt
2	14641	8172/9000	Network Status Interrupt
3	0	9000/9000	OIR interrupt
4	0	9000/9000	PCMCIA Interrupt
5	5849	8600/9000	Console Uart
6	0	9000/9000	Error Interrupt
7	396230	8604/9000	NMI Interrupt Handler

System was restarted by error - a Software forced crash, PC 0x602DE884 at 05:07:31

UTC Thu Sep 16 1999

RSP Software (RSP-JSV-M), Version 12.0(7)T, RELEASE SOFTWARE (fc2)

Compiled Mon 06-Dec-99 19:40 by phanguye

Image text-base: 0x60010908, database: 0x61356000

Stack trace from system failure:

FP: 0x61F73C30, RA: 0x602DE884

FP: 0x61F73C30, RA: 0x6030D29C

FP: 0x61F73D88, RA: 0x6025E96C

FP: 0x61F73DD0, RA: 0x6026A954

FP: 0x61F73E30, RA: 0x602B94BC

FP: 0x61F73E48, RA: 0x602B94A8

When a crashinfo is available in bootflash, this information is displayed at the end of the **show stacks** command:

```
*****
***** Information of Last System Crash *****
*****
```

Using bootflash:crashinfo_20000323-061850. 2000

CMD: 'sh int fas' 03:23:41 UTC Thu Mar 2 2000

CMD: 'sh int fastEthernet 6/0/0' 03:23:44 UTC Thu Mar 2 2000

CMD: 'conf t' 03:23:56 UTC Thu Mar 2 2000

CMD: 'no ip cef di' 03:23:58 UTC Thu Mar 2 2000

CMD: 'no ip cef distributed ' 03:23:58 UTC Thu Mar 2 2000

...

Router#**show context**

System was restarted by error - a Software forced crash, PC 0x602DE884 at

05:07:31 UTC Thu Sep 16 1999

RSP Software (RSP-JSV-M), Version 12.0(7)T, RELEASE SOFTWARE (fc2)

Compiled Mon 06-DEC-99 19:40 by phanguye

Image text-base: 0x60010908, database: 0x61356000

Stack trace from system failure:

FP: 0x61F73C30, RA: 0x602DE884

FP: 0x61F73C30, RA: 0x6030D29C

FP: 0x61F73D88, RA: 0x6025E96C

FP: 0x61F73DD0, RA: 0x6026A954

FP: 0x61F73E30, RA: 0x602B94BC

FP: 0x61F73E48, RA: 0x602B94A8

Fault History Buffer:

RSP Software (RSP-JSV-M), Version 12.0(7)T, RELEASE SOFTWARE (fc2)

Compiled Mon 06-DEC-99 19:40 by phanguye

Signal = 23, Code = 0x24, Uptime 3w0d

\$0 : 00000000, AT : 619A0000, v0 : 61990000, v1 : 00000032

a0 : 6026A114, a1 : 61A309A4, a2 : 00000000, a3 : 00000000

```
t0 : 61F6CD80, t1 : 8000FD88, t2 : 34008700, t3 : FFFF00FF
t4 : 00000083, t5 : 3E840024, t6 : 00000000, t7 : 00000000
s0 : 0000003C, s1 : 00000036, s2 : 00000000, s3 : 61F73C48
s4 : 00000000, s5 : 61993A10, s6 : 61982D00, s7 : 61820000
t8 : 0000327A, t9 : 00000000, k0 : 61E48C4C, k1 : 602E7748
gp : 6186F3A0, sp : 61F73C30, s8 : 00000000, ra : 6030D29C
EPC : 602DE884, SREG : 3400E703, Cause : 00000024
Error EPC : BFC00000, BadVaddr : 40231FFE
```

Information to Collect if You Open a TAC Service Request

If you still need assistance after you complete the troubleshooting steps, and want to open a service request with the Cisco TAC, be sure to include this information for troubleshooting a router crash:

- Troubleshooting performed before opening the service request.
- **show technical-support** output (if possible, in enable mode).
- **show log** output or console captures, if available.
- [crashinfo file](#) (if present, and not already included in the **show technical-support** output).
- **show region** output (if not already included in the **show technical-support** output).

Attach the collected data to your service request in non-zipped, plain text format (.txt). If you cannot access the Service Request tool, you can attach the relevant information to your service request by sending it to attach@cisco.com with your case number in the subject line of your message.

Note: Do not manually reload or power-cycle the router before you collect the information unless required to troubleshoot a router crash. This can cause important information to be lost that is needed to determine the root cause of the problem.