

Creating Core Dumps

When a router crashes, it can be useful to obtain a full copy of the memory image (core dump) to analyze the cause of the crash. Core dumps generally are only useful to your router technical support representative.



Caution Use the commands discussed in this appendix only in coordination with a technical support representative. The resulting binary file must be directed to a specific Trivial File Transfer Protocol (TFTP) server and subsequently interpreted by technical personnel who have access to source code and detailed memory maps.

Exception Commands

The **exception** class of configuration commands should be used only after consulting with a technical support representative. These commands are useful for debugging purposes, but they can result in unexpected behavior.

Creating a Core Dump

To obtain a core dump when a router crashes, use the **exception dump** *IP-address* router configuration command. *IP-address* is the address of your TFTP server. The core dump is written to a file named *hostname-core* on your TFTP server, where *hostname* is the name of the router (assigned using the **hostname** router configuration command). Using this command causes the router to attempt to make a core dump when it crashes.

This procedure cannot be guaranteed to work. It can fail for certain classes of system crashes. If successful, the core dump file will be the size of the memory available on the processor (for example, 4 MB for a CSC/3).

In addition you can use the **exception core-file** *filename* command to create a name for the core dump on your host.

Creating an Exception Memory Core Dump

During the debugging process, you can cause the router to create a core dump and reboot when certain memory size parameters are violated. The **exception memory** commands define a minimum contiguous block of memory in the free pool and a minimum size for the free memory pool.

[no] **exception memory fragment** *size*
[no] **exception memory minimum** *size*

The value of *size* is in bytes and is checked every 60 seconds. If you enter a size that is greater than the free memory, a core dump and router reload is generated after 60 seconds only when the **exception dump** command has been configured; otherwise, the router reloads without generating a core dump. The following example configures the router to monitor the free memory. If it falls below 250,000 bytes, it will dump the core and reload.

```
exception dump 131.108.92.2
exception core-file memory.overrun
exception memory minimum 250000
```

write core Command

You can test core dumps by using the EXEC command **write core**. This command causes the router to generate a dump without reloading and is useful if the router is malfunctioning, but has not crashed.

Depending on your TFTP server, you may need to create a target file before the router can write to it. You can test whether a target file is needed by attempting to use the TFTP **put** command from a workstation.

show Commands

When a router fails with an unexpected reload, and you report the problem to a technical support representative, always include a copy of the output from the **show stacks** and **show version** EXEC commands, so the representative can learn as much information about the state of your router when it failed.

show stacks Command

A useful EXEC command is **show stacks**. This command displays data saved by the ROM monitor, which includes a failure type, an operand address, and a failure program counter. This data is overwritten when the system is reloaded, so check your configuration register settings and decide how you want to recover from system crashes.

The “Memory Maps” appendix provides an example of **show stacks** output and memory map information that can help you determine whether a system crash is due to a software or hardware problem.

Software Version Identification

Much useful information is contained in the output of the **show version** command. (See Figure C-1.) The image type, version number, and function sets included in the image pinpoint the exact software that is running on your router.

Figure C-1 show version Command Output

```

500csl>show version
CS Software (CS500-K), Version 9.21(1.4)
Copyright (c) 1986-1993 by cisco Systems, Inc.
Compiled Fri 30-Apr-93 02:40 by jyang

ROM: System Bootstrap, Version 4.7(0.3)

```

Image type and function code	Version number
-------------------------------------	-----------------------

```

Cisco-500 (68331) processor with 10240K bytes of memory.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
1 Ethernet/IEEE 802.3 interface.
16 terminal lines.
32K bytes of non-volatile configuration memory.
Configuration register is 0x101

```

S2529

Version Numbering

A version number consists of a major version part, a minor version part, and an edit level, written in the form <major>.<minor>(<edit>). For Software Release 9.1(5), the major version is 9; the minor version is 1; and the edit level is 5. The major version number is changed when very extensive changes are made to the software. The minor number is changed whenever new functionality is added; and the edit level is changed when any change is made to the software.

For some versions, the edit level might also be divided into parts, as in Software Release 9.1(3.5). This convention identifies interim versions created between software maintenance releases. Software Release 9.1(3.5) would be the fifth 9.1 interim version created after Release 9.1(3). When a full-fledged field release in this progression is made, it would be numbered 9.1(4). Interim releases are distributed for special situations only, so most users see integer edit levels only.

Maintenance and interim releases progress linearly; 9.1(4) contains all the changes that were in Software Release 9.1(3) and all changes introduced in 9.1(3.x) versions. This linear progression is guaranteed for maintenance and interim releases, but not for major releases. For example, 9.17 does not contain all the functionality of 9.14; they are two separate products.

Beta test releases have two-part edit levels with zero as the first part; Software Release 9.1(0.11), for example, is a beta test release of 9.1. The first full field release of a software version always has edit level 1; for example, the first full field release of 9.1 is 9.1(1).

Experimental or early field test software may not follow edit-level numbering conventions. Such software often has single-number edit levels greater than 100, as in Software Release 9.1(12345). Also common are edit levels with more than two parts, edit levels that have nonnumeric components, and identifiers in brackets following the edit level. If you work with such software, pay careful attention to everything displayed by the **show version** command.

Image Types

The image type indicates which hardware platforms the image supports and the functions present in it, as shown in Table C-1. The version number indicates the source code version used to create the image. In addition, the version number indicates the release level of the image.

Table C-1 Image Types

Image Type	Description
GS7	Cisco 7000 series router software (GS stands for “Gateway Server” and is used for historical reasons).
GS2	Router software that runs on CSC/2 processors. Later GS2 images also will run on CSC/3 processors.
GS3	Software that runs on AGS+, AGS, MGS, and CGS routers with CSC/3 and CSC/4 processors.
XX	Cisco 4000 series router software (XX comes from the internal project code name used during Cisco 4000 development).
IGS	IGS and Cisco 3000 series router software and software for hub-based router products.
TS2	Older communication server (terminal server) software for CSC/2 processors.
TS3	Older communication server (terminal server) software for CSC/3 and CSC/4 processors.
CS3	Newer communication server software for CSC/3 and CSC/4 processors.
CS500	500-CS communication server software.
PT2	Protocol translator software for CSC/2 processors. Protocol translation is now available for many router and communication server products. In those products, the feature code P is used to denote the presence of protocol translation.
PT3	Protocol translator software for CSC/3 and CSC/4 processors.
STS	STS-10 terminal server software (the STS-10 is now obsolete).
BT2	Secondary bootstrap images for CSC/2 processors. Now largely obsolete.

Function Codes

Table C-2 describes function codes. Codes are often combined for images that have multiple optional function sets; for example, XX-RXBOOT is a Cisco 4000 image that executes from ROM, includes support for WAN connections, and includes only functions that are necessary for bootstrapping.

Table C-2 Function Types

Function Type	Description
B	Bridging support.
D	DDN X.25 support in very old images. Newer images include DDN X.25 with general X.25 support.
F	Full support. Used for older AGS+ images to denote the presence of support for the ciscoBus complex. Also used on Cisco 2000 series and Cisco 3000 series platforms to denote images that execute from Flash memory.
K	K images support all optional features (except for protocol translation) applicable to the platforms they run on. Note that even if all features are compiled into an image, most versions may require special licensing steps to enable the use of optional features.
P	Telnet/LAT/X.29 protocol translation.
R	Software that executes directly from ROM. R images cannot be network booted on any platform.
S	Standard software images without bridging, X.25, or other optional functionality.
X	X.25 and other WAN protocols.
BOOT	Minimal bootstrap images whose only function is to allow loading of other images over the network or from Flash memory.