Controller Unresponsiveness

- Upload Logs and Crash Files, on page 1
- Uploading Core Dumps from the Controller, on page 3
- Uploading Crash Packet Capture Files, on page 6
- Monitoring Memory Leaks, on page 9

Upload Logs and Crash Files

- Follow the instructions in this section to upload logs and crash files from the controller. However, before you begin, ensure you have a TFTP or FTP server available for the file upload. Follow these guidelines when setting up a TFTP or FTP server:
  - If you are uploading through the service port, the TFTP or FTP server must be on the same subnet as the service port because the service port is not routable, or you must create static routes on the controller.
  - If you are uploading through the distribution system network port, the TFTP or FTP server can be on the same or a different subnet because the distribution system port is routable.
  - A third-party TFTP or FTP server cannot run on the same computer as Cisco Prime Infrastructure because the Prime Infrastructure built-in TFTP or FTP server and the third-party TFTP or FTP server require the same communication port.

This section contains the following subsections:

Uploading Logs and Crash Files (GUI)

Procedure

Step 1 Choose **Command** > **Upload File**. The Upload File from Controller page appears.

Step 2 From the **File Type** drop-down list, choose one of the following:

- **Event Log**
- **Message Log**
- **Trap Log**

- **Crash File**

Step 3 From the **Transfer Mode** drop-down list, choose from the following options:

- **TFTP**
- **FTP**
- **SFTP**

Step 4 In the **IP Address** text box, enter the IP address of the server.

Step 5 In the **File Path** text box, enter the directory path of the log or crash file.

Step 6 In the **File Name** text box, enter the name of the log or crash file.

Step 7 If you chose FTP as the Transfer Mode, follow these steps:

a. In the **Server Login Username** text box, enter the FTP server login name.

b. In the **Server Login Password** text box, enter the FTP server login password.

c. In the **Server Port Number** text box, enter the port number of the FTP server. The default value for the server port is 21.

Step 8 Click **Upload** to upload the log or crash file from the controller. A message appears indicating the status of the upload.

Uploading Logs and Crash Files (CLI)

Procedure

Step 1 To transfer the file from the controller to a server, enter this command:

transfer upload mode {**tftp** | **ftp** | **sftp**}

Step 2 To specify the type of file to be uploaded, enter this command:

transfer upload datatype *datatype*

where *datatype* is one of the following options:

- **crashfile**—Uploads the system's crash file.
- **errorlog**—Uploads the system's error log.
- **panic-crash-file**—Uploads the kernel panic information if a kernel panic occurs.
- **systemtrace**—Uploads the system's trace file.
- **traplog**—Uploads the system's trap log.
- **watchdog-crash-file**—Uploads the console dump resulting from a software-watchdog-initiated reboot of the controller following a crash. The software watchdog module periodically checks the integrity of the internal software and makes sure that the system does not stay in an inconsistent or nonoperational state for a long period of time.

Step 3 To specify the path to the file, enter these commands:

- **transfer upload serverip** *server_ip_address*
- **transfer upload path** *server_path_to_file*
- **transfer upload filename** *filename*

Step 4 If you are using an FTP server, also enter these commands:

- **transfer upload username** *username*
- **transfer upload password** *password*
- **transfer upload port** *port*

Note
The default value for the port parameter is 21.

Step 5 To see the updated settings, enter this command:

transfer upload start

Step 6 When prompted to confirm the current settings and start the software upload, answer **y**.

Uploading Core Dumps from the Controller

To help troubleshoot controller crashes, you can configure the controller to automatically upload its core dump file to an FTP server after experiencing a crash. However, you cannot automatically send crash files to an FTP server.

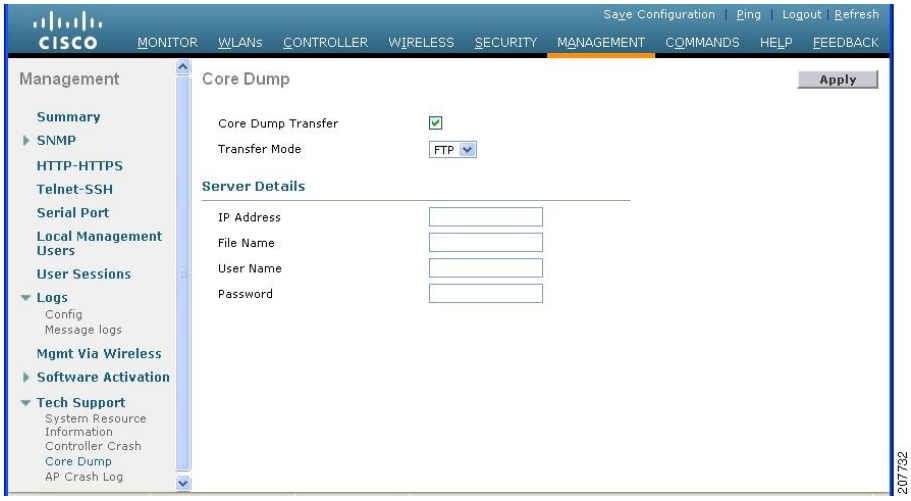
This section contains the following subsections:

Configuring the Controller to Automatically Upload Core Dumps to an FTP Server (GUI)

Procedure

Step 1 Choose **Management** > **Tech Support** > **Core Dump** to open the Core Dump page.

Figure 1: Core Dump Page



Step 2 To enable the controller to generate a core dump file following a crash, select the **Core Dump Transfer** check box.

Step 3 To specify the type of server to which the core dump file is uploaded, choose **FTP** from the **Transfer Mode** drop-down list.

Step 4 In the **IP Address** text box, enter the IP address of the FTP server.

Note
The controller must be able to reach the FTP server.

Step 5 In the **File Name** text box, enter the name that the controller uses to label the core dump file.

Step 6 In the **User Name** text box, enter the username for FTP login.

Step 7 In the **Password** text box, enter the password for FTP login.

Step 8 Click **Apply** to commit your changes.

Step 9 Click **Save Configuration** to save your changes.

Configuring the Controller to Automatically Upload Core Dumps to an FTP Server (CLI)

Procedure

Step 1 To enable or disable the controller to generate a core dump file following a crash, enter this command:

config coredump {**enable** | **disable**}

Step 2 To specify the FTP server to which the core dump file is uploaded, enter this command:

config coredump ftp *server_ip_address filename*

where

- *server_ip_address* is the IP address of the FTP server to which the controller sends its core dump file.

  Note
  The controller must be able to reach the FTP server.

- *filename* is the name that the controller uses to label the core dump file.

Step 3 To specify the username and password for FTP login, enter this command:
config coredump username *ftp_username* **password** *ftp_password*

Step 4 To save your changes, enter this command:
save config

Step 5 To see a summary of the controller's core dump file, enter this command:
show coredump summary

Example:
Information similar to the following appears:

```
Core Dump is enabled

FTP Server IP................................... 10.10.10.17
FTP Filename.................................... file1
FTP Username.................................... ftpuser
FTP Password.................................... *********
```

Uploading Core Dumps from Controller to a Server (CLI)

Procedure

Step 1 To see information about the core dump file in flash memory, enter this command:
show coredump summary
Information similar to the following appears:

```
Core Dump is disabled

Core Dump file is saved on flash

Sw Version..................................... 6.0.83.0
Time Stamp..................................... Wed Feb  4 13:23:11 2009
File Size...................................... 9081788
File Name Suffix.......................... filename.gz
```

Step 2 To transfer the file from the controller to a server, enter these commands:

- **transfer upload mode** {**tftp** | **ftp** | **sftp**}
- **transfer upload datatype coredump**

- **transfer upload serverip** *server_ip_address*
- **transfer upload path** *server_path_to_file*
- **transfer upload filename** *filename*

Note
After the file is uploaded, it ends with a .gz suffix. If desired, you can upload the same core dump file multiple times with different names to different servers.

Step 3 If you are using an FTP server, also enter these commands:

- **transfer upload username** *username*

  transfer upload password *password*

- **transfer upload port** *port*

Note
The default value for the *port* parameter is 21.

Step 4 To view the updated settings, enter this command:

transfer upload start

Step 5 When prompted to confirm the current settings and start the software upload, answer y.

Uploading Crash Packet Capture Files

When a controller's data plane crashes, it stores the last 50 packets that the controller received in flash memory. This information can be useful in troubleshooting the crash.

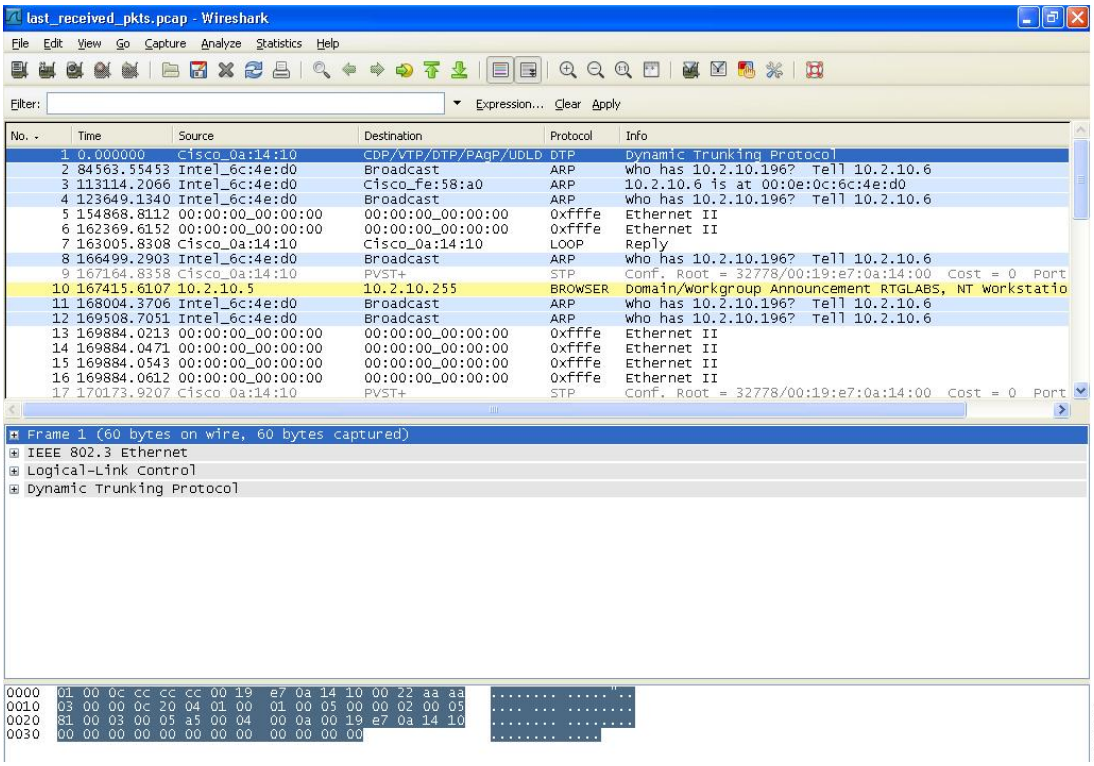
When a crash occurs, the controller generates a new packet capture file (*.pcap) file, and a message similar to the following appears in the controller crash file:

```
Last 5 packets processed at each core are stored in
 "last_received_pkts.pcap" captured file.
   - Frame 36,38,43,47,49, processed at core #0.
   - Frame 14,27,30,42,45, processed at core #1.
   - Frame 15,18,20,32,48, processed at core #2.
   - Frame 11,29,34,37,46, processed at core #3.
   - Frame 7,8,12,31,35, processed at core #4.
   - Frame 21,25,39,41,50, processed at core #5.
   - Frame 16,17,19,22,33, processed at core #6.
   - Frame 6,10,13,23,26, processed at core #7.
   - Frame 9,24,28,40,44, processed at core #8.
   - Frame 1,2,3,4,5, processed at core #9.
```

You can use the controller GUI or CLI to upload the packet capture file from the controller. You can then use Wireshark or another standard packet capture tool to view and analyze the contents of the file.

Figure 2: Sample Output of Packet Capture File in Wireshark

This figure shows a sample output of the packet capture in Wireshark.



This section contains the following subsections:

Restrictions for Uploading Crash Packet Capture Files

- Ensure that you have a TFTP or FTP server available for the file upload. Follow these guidelines when setting up a TFTP or FTP server:
  - If you are uploading through the service port, the TFTP or FTP server must be on the same subnet as the service port because the service port is not routable, or you must create static routes on the controller.
  - If you are uploading through the distribution system network port, the TFTP or FTP server can be on the same or a different subnet because the distribution system port is routable.
  - A third-party TFTP or FTP server cannot run on the same computer as Cisco Prime Infrastructure because the Prime Infrastructure built-in TFTP or FTP server and the third-party TFTP or FTP server require the same communication port.

Uploading Crash Packet Capture Files (GUI)

Procedure

Step 1 Choose **Commands** > **Upload File** to open the **Upload File from Controller** page.

Step 2 From the **File Type** drop-down list, choose **Packet Capture**.

Step 3 From the **Transfer Mode** drop-down list, choose from the following options:

- **TFTP**
- **FTP**
- **SFTP**

Step 4 In the **IP Address** field, enter the IP address of the server.

Step 5 In the **File Path** field, enter the directory path of the packet capture file.

Step 6 In the **File Name** field, enter the name of the packet capture file. These files have a .pcap extension.

Step 7 If you are using an FTP server, follow these steps:

a) In the **Server Login Username** field, enter the username to log into the FTP server.
b) In the **Server Login Password** field, enter the password to log into the FTP server.
c) In the **Server Port Number** field, enter the port number on the FTP server through which the upload occurs. The default value is 21.

Step 8 Click **Upload** to upload the packet capture file from the controller. A message is displayed indicating the status of the upload.

Step 9 Use Wireshark or another standard packet capture tool to open the packet capture file and see the last 50 packets that were received by the controller.

Uploading Crash Packet Capture Files (CLI)

Procedure

Step 1 Log on to the controller CLI.

Step 2 Enter the **transfer upload mode** {**tftp** | **ftp** | **sftp**} command.

Step 3 Enter the **transfer upload datatype packet-capture** command.

Step 4 Enter the **transfer upload serverip** *server-ip-address* command.

Step 5 Enter the **transfer upload path** *server-path-to-file* command.

Step 6 Enter the **transfer upload filename** *last_received_pkts*.pcap command.

Step 7 If you are using an FTP server, enter these commands:

- **transfer upload username** *username*
- **transfer upload password** *password*
- **transfer upload port** *port*

Note
The default value for the *port* parameter is 21.

Step 8 Enter the **transfer upload start** command to see the updated settings and then answer **y** when prompted to confirm the current settings and start the upload process.

Step 9 Use Wireshark or another standard packet capture tool to open the packet capture file and see the last 50 packets that were received by the controller.

Monitoring Memory Leaks

This section provides instructions for troubleshooting hard-to-solve or hard-to-reproduce memory problems.

Caution The commands in this section can be disruptive to your system and should be run only when you are advised to do so by the Cisco Technical Assistance Center (TAC).

This section contains the following subsection:

Monitoring Memory Leaks (CLI)

Procedure

Step 1 To enable or disable monitoring for memory errors and leaks, enter this command:

config memory monitor errors {**enable** | **disable**}

The default value is disabled.

Note
Your changes are not saved across reboots. After the controller reboots, it uses the default setting for this feature.

Step 2 If you suspect that a memory leak has occurred, enter this command to configure the controller to perform an auto-leak analysis between two memory thresholds (in kilobytes):

config memory monitor leaks *low_thresh high_thresh*

If the free memory is lower than the *low_thresh* threshold, the system crashes, generating a crash file. The default value for this parameter is 10000 kilobytes, and you cannot set it below this value.

Set the *high_thresh* threshold to the current free memory level or higher so that the system enters auto-leak-analysis mode. After the free memory reaches a level lower than the specified *high_thresh* threshold, the process of tracking and freeing memory allocation begins. As a result, the **debug memory events enable** command shows all allocations and frees, and the **show memory monitor detail** command starts to detect any suspected memory leaks. The default value for this parameter is 30000 kilobytes.

Step 3 To see a summary of any discovered memory issues, enter this command:

show memory monitor

Information similar to the following appears:

```
Memory Leak Monitor Status:
low_threshold(10000), high_threshold(30000), current status(disabled)
```

```
-----------------------------------------

Memory Error Monitor Status:
Crash-on-error flag currently set to (disabled)
No memory error detected.
```

Step 4 To see the details of any memory leaks or corruption, enter this command:

show memory monitor detail

Information similar to the following appears:

```
Memory error detected. Details:
-----------------------------------------------
-  Corruption detected at pmalloc entry address:        (0x179a7ec0)
-  Corrupt entry:headerMagic(0xdeadf00d),trailer(0xabcd),poison(0xreadceef),
entrysize(128),bytes(100),thread(Unknown task name, task id = (332096592)),
file(pmalloc.c),line(1736),time(1027)

Previous 1K memory dump from error location.
-----------------------------------------------
(179a7ac0): 00000000 00000000 00000000 ceeff00d readf00d 00000080 00000000 00000000
(179a7ae0): 17958b20 00000000 1175608c 00000078 00000000 readceef 179a7afc 00000001
(179a7b00): 00000003 00000006 00000001 00000004 00000001 00000009 00000009 0000020d
(179a7b20): 00000001 00000002 00000002 00000001 00000004 00000000 00000000 5d7b9aba
(179a7b40): cbddf004 192f465e 7791acc8 e5032242 5365788c a1b7cee6 00000000 00000000
(179a7b60): 00000000 00000000 00000000 00000000 00000000 ceeff00d readf00d 00000080
(179a7b80): 00000000 00000000 17958dc0 00000000 1175608c 00000078 00000000 readceef
(179a7ba0): 179a7ba4 00000001 00000003 00000006 00000001 00000004 00000001 00003763
(179a7bc0): 00000002 00000002 00000010 00000001 00000002 00000000 0000001e 00000013
(179a7be0): 0000001a 00000089 00000000 00000000 000000d8 00000000 00000000 17222194
(179a7c00): 1722246c 1722246c 00000000 00000000 00000000 00000000 00000000 ceeff00d
(179a7c20): readf00d 00000080 00000000 00000000 179a7b78 00000000 1175608c 00000078
```

Step 5 If a memory leak occurs, enter this command to enable debugging of errors or events during memory allocation:
debug memory {errors | events} {enable | disable}

Troubleshooting Memory Leaks

To investigate the cause for low memory state, follow these steps:

Procedure

Step 1 **show memory statistics**

Step 2 **test system cat** */proc/meminfo*

Step 3 **show system top**

```
PID
1078 root      18   0  4488  888  756 S    0  0.1   0:00.00 gettyOrMwar
1081 root      20   0  980m 557m  24m S    0 56.9  41:33.32 switchdrvr
```

In this example, the PID to focus on is 1081.

Step 4 **test system cat** */proc/1081/smaps*

Step 5 **show system timers ticks-exhausted**

```
Timer Ticks ..................................... 3895180 ticks   (779036 seconds)
```

Here focus on the seconds value 779036.

Step 6 **show memory allocations [all/<pid>] [all/<pool-size>] [<start_time>] [<end_time>]**

If you see any allocations, they are probable memory leak candidates. You need to check if these are valid allocations made earlier to the low memory state issue.