

Chapter 24

Collect Crash Data

This chapter explains the crashes that can occur in different areas of the JUNOS software, and provides procedures you use to collect the crash data necessary for troubleshooting by the Juniper Networks Technical Assistance Center (JTAC). (See Table 58.)

Table 58: Checklist for Collecting Crash Data

Collecting Crash Data Tasks	Command or Action
Understand Crash Data Collection on page 301	
Collect Crash Data for a Routing Engine Kernel on page 302	
1. Check the Routing Engine Core Files on page 302	file list detail /var/crash
2. Collect and Send Crash Data to JTAC on page 303	
a. List the Core Files on page 303	start shell su root password cd /var/crash ls -l
b. Compress the vmcore File on page 304	gzip vmcore.number To unzip the vmcore file: gzip -d vmcore.number.gz
c. Log Software Version Information on page 304	show version
d. Open a Case with JTAC on page 305	support@juniper.net ftp ftp.juniper.net
Collect Crash Data for Routing Engine Daemons on page 306	
1. Check for Daemon Core Files on page 306	file list detail /var/tmp
2. Collect and Send Crash Data to JTAC on page 307	
a. List the Daemon Core Files on page 307	start shell su root password cd /var/tmp ls -l
b. Compress the Daemon Core Files on page 308	gzip daemon-executable-name.core.number
c. Log Software Version Information on page 308	show version
d. Open a Case with JTAC on page 309	support@juniper.net ftp ftp.juniper.net

Collecting Crash Data Tasks	Command or Action
Collect Crash Data for the Packet Forwarding Engine Microkernel on page 310	
1. Display the Crash Stack Traceback and Registration Information on page 311	start shell su root password vty <i>component-executable-name</i> show nvram show syslog messages
2. Clear the NVRAM Contents on page 314	start shell su root password vty <i>component-executable-name</i> clear nvram
3. Check Packet Forwarding Engine Microkernel Core Files on page 315	file list detail /var/crash
4. Collect and Send Crash Data to JTAC on page 315	
a. List the Core Files Generated by the Crash on page 315	start shell su root password cd /var/crash ls -l
b. Compress the Core Files on page 316	gzip <i>filename</i>
c. Log Software Version Information on page 316	show version
d. Open a Case with JTAC on page 317	support@juniper.net ftp ftp.juniper.net

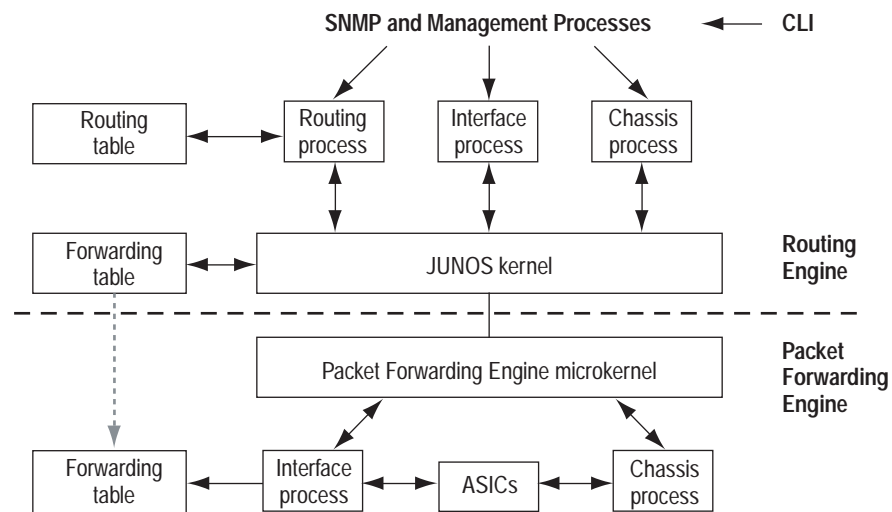
Understand Crash Data Collection

A crash can occur in one of three areas in the JUNOS software architecture (see Figure 22):

- Routing Engine kernel
- Routing Engine daemons (processes)
- Packet Forwarding Engine microkernel

Figure 22: Three Areas Where a Software Crash Can Occur

Software Architecture



1992

Collect Crash Data for a Routing Engine Kernel

Purpose When a Routing Engine kernel crashes, the Routing Engine automatically reboots. By default, the Juniper Networks router does not attempt to dump a core if the Routing Engine kernel crashes. As a result, there is no crash data on the router to help investigate the crash. In addition, the system log messages are similar to those generated when the router is powered down and restarted, so you cannot tell if the Routing Engine restart was caused by a kernel crash or a normal power restart.

Steps To Take To collect crash data for a Routing Engine kernel crash, follow these steps:

1. Check the Routing Engine Core Files on page 302
2. Collect and Send Crash Data to JTAC on page 303

Step 1: Check the Routing Engine Core Files

Purpose If you observe disruption to the Routing Engine kernel, check the `/var/crash` directory for any core files created around the time of the crash.

Action To check the `/var/crash` directory, use the following JUNOS command-line interface (CLI) operational mode command:

```
user@host> file list detail /var/crash
```

Sample Output

```
user@host> file list detail /var/crash
total 1577912
drwxr-x---  2 root  wheel           512 Sep  9 11:59 ./
drwxr-xr-x 22 root  wheel           512 Oct 29 2001 ../
-rw-r--r--  1 root  wheel             2 Jul 20 01:11 bounds
-rw-r--r--  1 root  wheel      2166913 Jul 20 01:11 kernel.0
-rw-r--r--  1 root  wheel             5 Feb 15 2002 minfree
-rw-----  1 root  wheel    805306368 Jul 20 01:11 vmcore.0
```

What It Means The sample output lists the contents of the `/var/crash/` directory. Check the date and timestamp for any kernel core files created around the time of the crash. In the example above, two core files are listed: `kernel.0` and `vmcore.0`.

Step 2: Collect and Send Crash Data to JTAC

Purpose If a Routing Engine kernel crash occurs on your router, collect the following data for JTAC evaluation and instruction.

Steps To Take To collect and send crash data to JTAC, follow these steps:

1. List the Core Files on page 303
2. Compress the vmcore File on page 304
3. Log Software Version Information on page 304
4. Open a Case with JTAC on page 305

List the Core Files

Action To list the core files, follow these steps:

1. Exit from the CLI environment and create a UNIX-level shell by entering the `start shell` command:

```
user@host> start shell
```

2. Type `su` and the root password when prompted. You are now in the shell and the prompt is `%` instead of `>`, for example:

```
% su
Password: ****
```

3. Change the directory to `/var/crash` and type `ls -l`, for example:

```
root@host% cd /var/crash
root@host% ls -l
```

4. Look for any core files created around the time of the crash.

Sample Output

```
user@host> start shell
% su
Password: ****
root@host% cd /var/crash
root@host% ls -l
total 1577908
-rw-r--r--  1 root  wheel           2 Jul 20 01:11 bounds
-rw-r--r--  1 root  wheel    2166913 Jul 20 01:11 kernel.0
-rw-r--r--  1 root  wheel         5 Feb 15  2002 minfree
-rw-----  1 root  wheel  805306368 Jul 20 01:11 vmcore.0
```

What It Means The sample output lists the contents of the `/var/crash` directory and shows the current core files `kernel.0` and `vm.core.0`.

Compress the vmcore File

Purpose gzip is a compression utility used to compress files. Compress the `vmcore` file if it is larger than 50 MB. Files created using the `gzip` command end with the file extension `.gz`.



NOTE: Use lowercase for the `gzip` command when you are in the shell.

Action To compress the `vmcore` file with `gzip`, use the following command from the shell:

```
root@host% gzip vmcore.number
```

To unzip the `vmcore` file with `gzip`, use the following command from the shell:

```
root@host% gzip -d vmcore.number.gz
```

What It Means The contents of the `vmcore` file are compressed into a single compressed file named `vmcore.number.gz`. The `gzip` command preserves the mode, ownership, and timestamps of files when compressing or decompressing them.

Log Software Version Information

Action To log the JUNOS software version information, use the following JUNOS CLI operational mode command:

```
user@host> show version
```

Sample Output user@host> `show version`

```
Hostname: host
Model: m10
JUNOS Base OS boot [5.0R5]
JUNOS Base OS Software Suite [5.0R5]
JUNOS Kernel Software Suite [5.0R5]
JUNOS Routing Software Suite [5.0R5]
JUNOS Packet Forwarding Engine Support [5.0R5]
JUNOS Crypto Software Suite [5.0R5]
JUNOS Online Documentation [5.0R5]
KERNEL 5.0R5 #0 built by builder on 2002-03-02 05:10:28 UTC
MGD release 5.0R5 built by builder on 2002-03-02 04:45:32 UTC
CLI release 5.0R5 built by builder on 2002-03-02 04:44:22 UTC
CHASSISD release 5.0R5 built by builder on 2002-03-02 04:43:37 UTC
DCD release 5.0R5 built by builder on 2002-03-02 04:42:47 UTC
RPD release 5.0R5 built by builder on 2002-03-02 04:46:17 UTC
SNMPD release 5.0R5 built by builder on 2002-03-02 04:52:26 UTC
MIB2D release 5.0R5 built by builder on 2002-03-02 04:45:37 UTC
APSD release 5.0R5 built by builder on 2002-03-02 04:43:31 UTC
VRRPD release 5.0R5 built by builder on 2002-03-02 04:52:34 UTC
ALARM release 5.0R5 built by builder on 2002-03-02 04:43:24 UTC
PFED release 5.0R5 built by builder on 2002-03-02 04:46:06 UTC
CRAFTD release 5.0R5 built by builder on 2002-03-02 04:44:30 UTC
SAMPLED release 5.0R5 built by builder on 2002-03-02 04:52:20 UTC
ILMID release 5.0R5 built by builder on 2002-03-02 04:45:21 UTC
BPRELAYD release 5.0R5 built by builder on 2002-03-02 04:42:41 UTC
RMOPD release 5.0R5 built by builder on 2002-03-02 04:46:11 UTC
jkernel-dd release 5.0R5 built by builder on 2002-03-02 04:41:07 UTC
jroute-dd release 5.0R5 built by builder on 2002-03-02 04:41:21 UTC
```

```
jdocs-dd release 5.0R5 built by builder on 2002-03-02 04:39:11 UTC
```

What It Means The sample output shows the hostname, router model, and the different JUNOS software packages, processes, and documents.

Open a Case with JTAC

Action To open a case with JTAC, follow these steps:

1. Send an e-mail to support@juniper.net, and include the information from the `show version` command.
2. At the support engineer's request, ftp the `vmcore.number.gz` file to a case-number directory at [ftp.juniper.net](ftp://ftp.juniper.net). To ftp the core file to a directory, follow these steps:
 - a. At the shell prompt, enter `ftp ftp.juniper.net`.
 - b. At the name prompt, enter `anonymous`.
 - c. At the password prompt, enter your e-mail address as the password.
 - d. At the ftp prompt, enter the `cd pub/ incoming` string.
 - e. Enter the `mkdir case-number` command, where the *case-number* is the value of the case you opened with JTAC, for example, `1999-1231-9999`. If a directory has already been created, continue with the next step.
 - f. Enter the `cd case-number` command.
 - g. Enter the `binary` command so that the file transfer is in binary and not ASCII.
 - h. Enter the `put vmcore.0.gz` command.

Sample Output The following output is an example of copying a core file from the shell to an ftp directory at [ftp.juniper.net](ftp://ftp.juniper.net):

```
root@host% ftp ftp.juniper.net
Connected to colo-ftp.juniper.net.
220 colo-ftp.juniper.net FTP server (Version 6.00LS) ready.
Name (ftp.juniper.net: root): anonymous
331 Guest login ok, send your email address as password.
Password: ****
230 Guest login ok, access restrictions apply.
ftp> cd pub/ incoming
250 CWD command successful
ftp> mkdir 1999-1231-9999
257 MKD command successful.
ftp> cd 1999-1231-9999
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> put vmcore.0.gz
```

What It Means The sample output shows that there is a connection to `ftp.juniper.net`, that the login name and password were entered, and that the core file was successfully copied from the shell to an ftp directory at `ftp.juniper.net`.

Collect Crash Data for Routing Engine Daemons

Steps To Take To collect crash data for Routing Engine daemons, follow these steps:

1. Check for Daemon Core Files on page 306
2. Collect and Send Crash Data to JTAC on page 307

Step 1: Check for Daemon Core Files

Purpose If you observe disruption to routing protocol operation, system log operation, Simple Network Management Protocol (SNMP) operation, or other operations handled by Routing Engine daemons, check the `/var/tmp` directory for any daemon core files created around the time of the crash.

Action To check the `/var/tmp` directory, use the following JUNOS CLI operational mode command:

```
user@host> file list detail /var/tmp
```

Sample Output

```
user@host> file list detail /var/tmp
total 1292622
drwxrwxrwt  3 root  field      512 Dec 31 06:48 ./
drwxr-xr-x 21 root  field      512 Mar  5 1999 ../
-rw-rw----  1 root  field 119713792 Nov 17 21:58 rpd.core.0
-rw-rw----  1 root  field 120782848 Nov 17 22:12 rpd.core.1
```

What It Means The sample output lists the contents of the `/var/tmp/` directory. Look for any daemon core files created around the time of the crash. In the example above, two core files are listed: `rpdc.core.0` and `rpdc.core.1`.

Table 59 lists the major Routing Engine daemons supported by the JUNOS software.

Table 59: Major Routing Engine Daemons

Executable Name	Definition	Description
rpdc	Routing protocol daemon	Provides routing protocol intelligence (Border Gateway Protocol [BGP], Intermediate System-to-Intermediate System [ISIS], Open Shortest Path First [OSPF], and so on).
dcd	Device control daemon	Manages all interface devices.
mgd	Management daemon	Provides user configuration access to the system. The CLI is a client of mgd .
snmpd	Simple Network Management Protocol daemon	Provides remote network management information to the network management system.
chassisd	Chassis daemon	Monitors and manages Flexible PIC Concentrator (FPC) slots and other environmental components.
alarmd	Alarm daemon	Manages system alarm notifications.

Executable Name	Definition	Description
apsd	Automatic protection switching daemon	Provides SONET Automatic Protection Switching (APS) functionality.
sampled	Traffic sampling daemon	Gathers traffic sampling information.
vrrpd	Virtual Router Redundancy Protocol daemon	Provides Virtual Router Redundancy Protocol (VRRP) functionality.
syslogd	System log daemon	Manages the router system logging operation.
mib2d	MIB2 daemon	Management Information Base (MIB) subagent for MIB2.

Step 2: Collect and Send Crash Data to JTAC

Purpose If a Routing Engine daemon crash occurs on your router and a core file is created in the `/var/tmp` directory, collect the following data for JTAC evaluation and instruction.

Steps To Take To collect and send crash data to JTAC, follow these steps:

1. List the Daemon Core Files on page 307
2. Compress the Daemon Core Files on page 308
3. Log Software Version Information on page 308
4. Open a Case with JTAC on page 309

List the Daemon Core Files

Action To list the daemon core files, follow these steps:

1. Exit from the CLI environment and create a UNIX-level shell by entering the `start shell` command:

```
user@host> start shell
```

2. Type `su` and the root password when prompted. You are now in the shell and the prompt is `%` instead of `>`, for example:

```
% su
Password: ****
```

3. Change the directory to `/var/tmp` and type `ls -l`, for example:

```
root@host% cd /var/tmp
root@host% ls -l
```

4. Look for any daemon core files created around the time of the crash.

Sample Output

```
user@host> start shell
% su
Password: ****
root@host% cd /var/tmp
root@host% ls -l
total 1292618
```

```
-rw-rw---- 1 root field 119713792 Nov 17 21:58 rpd.core.0
-rw-rw---- 1 root field 120782848 Nov 17 22:12 rpd.core.1
```

What It Means The sample output lists the contents of the `/var/tmp` directory and shows the current core file (`rpd.core.1`) and one previous core file (`rpd.core.0`) for the routing protocol daemon (`rpd`). For each daemon, you can have a total of five core files in the `/var/tmp` directory: the current core file and the four previous core files numbered 0 through 4 (from oldest to newest).

Compress the Daemon Core Files

Purpose `gzip` is a compression utility used to compress the files if they are large. Files created using the `gzip` command end with the file extension `.gz`. Compress the core file if it is over 50 MB.



NOTE: Use lowercase for the `gzip` command when you are in the shell.

You only need to compress the daemon core files when the tarball file is not created.

Action To compress the daemon core file with `gzip`, use the following command from the shell:

```
root@host% gzip daemon-executable-name.core.number
```

Sample Output `root@host% gzip rpd.core.0`

What It Means The contents of the daemon core file are compressed into a single compressed file named `daemon.number.gz`. The `gzip` command preserves the mode, ownership, and timestamps of files when compressing or decompressing them.

Log Software Version Information

Action To log JUNOS software version information, use the following JUNOS CLI operational mode command:

```
user@host> show version
```

Sample Output

```
user@host> show version
Hostname: host
Model: m10
JUNOS Base OS boot [5.0R5]
JUNOS Base OS Software Suite [5.0R5]
JUNOS Kernel Software Suite [5.0R5]
JUNOS Routing Software Suite [5.0R5]
JUNOS Packet Forwarding Engine Support [5.0R5]
JUNOS Crypto Software Suite [5.0R5]
JUNOS Online Documentation [5.0R5]
KERNEL 5.0R5 #0 built by builder on 2002-03-02 05:10:28 UTC
MGD release 5.0R5 built by builder on 2002-03-02 04:45:32 UTC
CLI release 5.0R5 built by builder on 2002-03-02 04:44:22 UTC
CHASSISD release 5.0R5 built by builder on 2002-03-02 04:43:37 UTC
DCD release 5.0R5 built by builder on 2002-03-02 04:42:47 UTC
RPD release 5.0R5 built by builder on 2002-03-02 04:46:17 UTC
SNMPD release 5.0R5 built by builder on 2002-03-02 04:52:26 UTC
```

```

MIB2D release 5.0R5 built by builder on 2002-03-02 04:45:37 UTC
APSD release 5.0R5 built by builder on 2002-03-02 04:43:31 UTC
VRRPD release 5.0R5 built by builder on 2002-03-02 04:52:34 UTC
ALARMD release 5.0R5 built by builder on 2002-03-02 04:43:24 UTC
PFED release 5.0R5 built by builder on 2002-03-02 04:46:06 UTC
CRAFTD release 5.0R5 built by builder on 2002-03-02 04:44:30 UTC
SAMPLED release 5.0R5 built by builder on 2002-03-02 04:52:20 UTC
ILMID release 5.0R5 built by builder on 2002-03-02 04:45:21 UTC
BPDELAYD release 5.0R5 built by builder on 2002-03-02 04:42:41 UTC
RMOPD release 5.0R5 built by builder on 2002-03-02 04:46:11 UTC
jkernel-dd release 5.0R5 built by builder on 2002-03-02 04:41:07 UTC
jroute-dd release 5.0R5 built by builder on 2002-03-02 04:41:21 UTC
jdocs-dd release 5.0R5 built by builder on 2002-03-02 04:39:11 UTC

```

What It Means The output shows the hostname, router model, and the different JUNOS software packages, processes, and documents.

Open a Case with JTAC

Action To open a case with JTAC, follow these steps:

1. Send an e-mail to support@juniper.net, and include the information from the `show version` command.
2. At the support engineer's request, ftp the `daemon.number.gz` file to a directory at `ftp.juniper.net`. To ftp the core file to a directory, follow these steps:
 - a. At the shell prompt, enter `ftp ftp.juniper.net`.
 - b. At the name prompt, enter `anonymous`.
 - c. At the password prompt, enter your e-mail address as the password.
 - d. At the ftp prompt, enter the `cd pub/ incoming` string.
 - e. Enter the `mkdir case-number` command, where the *case-number* is the value of the case you opened with JTAC, for example, `1999-1231-9999`. If a directory has already been created, continue with the next step.
 - f. Enter the `cd case-number` command.
 - g. Enter the `binary` command so that the file transfer is in binary and not ASCII.
 - h. Enter the `put daemon.core.number.gz` command.

Sample Output The following output is an example of copying a core file from the shell to an ftp directory at `ftp.juniper.net`:

```

root@host% ftp ftp.juniper.net
Connected to colo-ftp.juniper.net.
220 colo-ftp.juniper.net FTP server (Version 6.00LS) ready.
Name (ftp.juniper.net: root): anonymous
331 Guest login ok, send your email address as password.
Password: ****
230 Guest login ok, access restrictions apply.
ftp> cd pub/ incoming

```

```

250 CWD command successful
ftp> mkdir 1999-1231-9999
257 MKD command successful.
ftp> cd 1999-1231-9999
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> put rpd.4.gz

```

What It Means The sample output shows that there is a connection to `ftp.juniper.net`, that the login name and password were entered, and that the core file was successfully copied from the shell to an ftp directory at `ftp.juniper.net`.

Collect Crash Data for the Packet Forwarding Engine Microkernel

Purpose Each of the following Packet Forwarding Engine components of a Juniper Networks router runs a microkernel:

- Flexible PIC Concentrator (FPC) on M-series platforms except for the M5 and M10 Internet routers
- Gibson Flexible PIC Concentrator (GFPC) on T640 and T320 Internet routing nodes
- Switched Printed Mezzanine Board (SPMB) on T640 and T320 Internet routing nodes
- Forwarding Engine Board (FEB) on M5 and M10 Internet routers
- System Switching Board (SSB) on an M20 Internet router
- System Control Board (SCB) on an M40 Internet router
- Switching and Forwarding Module (SFM) on M160 and M40e Internet routers

When a crash occurs, crash stack traceback and registration information is placed into nonvolatile random access memory (NVRAM) on the different components. Table 60 shows where the NVRAM is located for the components for each router.

Table 60: NVRAM Location on the Microkernel of the Packet Forwarding Engine Components

Router Type	NVRAM Location
M5 and M10	FEB
M20	SSB and crash stack traceback and register information for the FPC
M40	SCB and crash stack traceback and register information for the FPC
M40e	FPC SFM
M160	FPC SFM

Router Type	NVRAM Location
T320	GFPC SPMB
T640	GFPC SPMB

Steps To Take To collect crash data for the Packet Forwarding Engine microkernel, follow these steps:

1. Display the Crash Stack Traceback and Registration Information on page 311
2. Clear the NVRAM Contents on page 314
3. Check Packet Forwarding Engine Microkernel Core Files on page 315
4. Collect and Send Crash Data to JTAC on page 315

Step 1: Display the Crash Stack Traceback and Registration Information

Action To display the crash stack traceback and registration information, follow these steps:

1. Exit from the CLI environment and create a UNIX-level shell by entering the `start shell` command:

```
user@host> start shell
```

2. Type `su` and the root password when prompted. You are now in the shell and prompt is `%` instead of `>`, for example:

```
% su
Password: ****
```

3. Establish a vty session to the appropriate component. Use the `vtty` command followed by the executable name for the component; for example, `scb`, `ssb0`, `ssb1`, `fpc0`, or `fpc1`:

```
root@host% vty sfm0
```



NOTE: For the M40e and M160 routers, you can also create a `cty` session to the components if the components are not online.

4. Type the `show nvram` command to view the NVRAM information.
5. Type the `show syslog messages` command to view the system log messages.

Sample Output 1

```
user@host> start shell
% su
Password: ****
root@host% vty sfm0
```

```
SFM platform (266Mhz PPC 603e processor, 64Mb memory, 512Kb flash)
```

```

SFM3(host vty)# show nvram
System NVRAM :
  4080 available bytes, 4080 used, 0 free
  Contents:

mpc106 machine check caused by error on the PCI Bus
mpc106 error detect register 1: 0x08, 2: 0x00
mpc106 error ack count = 0
mpc106 error address: 0x0a000000
mpc106 PCI bus error status register: 0x02
  mpc106 was the PCI master
  C/BE bits: I/O read [0b0010]
mpc106 error detection reg1: PCI cycle
mpc106 PCI status reg: parity error

System Exception: Vector/Code 0x00700, Signal 4
Event occurred at: Oct 26 13:32:40.952

Juniper Embedded Microkernel Version 4.2R1
Built by tlim on 2000-09-23 06:11:28 UTC
Copyright (C) 1998-2000, Juniper Networks, Inc.
All rights reserved.
Reason string: "Program Check"
Context: Thread (PFE Manager)

Registers:
R00: 0x06f5f81c R01: 0x06f5f9cc R02: 0x00003344 R03: 0x00000000
R04: 0x00008000 R05: 0x00000000 R06: 0x0010052c R07: 0x06f637e4
R08: 0x06f5f81c R09: 0x00169810 R10: 0x000000e8 R11: 0x00000001
R12: 0x00046cdf R13: 0xffffffff R14: 0xffffffff R15: 0xffffffff
R16: 0xffffffff R17: 0xffffffff R18: 0xffffffff R19: 0xffffffff
R20: 0xffffffff R21: 0xffffffff R22: 0xffffffff R23: 0xffffffff
R24: 0x00000003 R25: 0x00000000 R26: 0x00000001 R27: 0x0000fc78
R28: 0x00150000 R29: 0x0016c4b0 R30: 0x06f5eb7c R31: 0x97cb1d36
MSR: 0x0008b030 CTR: 0x000ac008 Link:0x06f5f81c SP: 0x06f5f9cc
CCR: 0x22200024 XER: 0x20000000 PC: 0x06f5f81c
DSISR: 0x00000000 DAR: 0xffffffff K_MSR: 0x00001030

Stack Traceback:
Frame 01: sp = 0x06f5f9cc, pc = 0x06f5f81c
Frame 02: sp = 0x06f5f9e4, pc = 0x000c7e28
Frame 03: sp = 0x06f5fa04, pc = 0x00026620

ROM NVRAM:
  0 available bytes, 0 used, 0 free

SFM3(host vty)# show syslog messages
Oct 26 12:02:05 router tnp_sfm_2 PFEMAN: sent Resync request to Master
Oct 26 12:02:07 router tnp_sfm_3 CM(3): Slot 1: On-line
Oct 26 12:02:07 router tnp_sfm_3 CM(3): Slot 2: On-line
Oct 26 12:02:07 router tnp_sfm_3 CM(3): Slot 6: On-line
Oct 26 12:02:07 router tnp_sfm_3 PFEMAN: sent Resync request to Master
Oct 26 12:05:58 router tnp_sfm_3 mpc106 machine check caused by error on the
PCI Bu
s
Oct 26 12:05:58 router tnp_sfm_3 mpc106 error detect register 1: 0x08,
2: 0x00
Oct 26 12:05:58 router tnp_sfm_3 mpc106 error ack count = 0
Oct 26 12:05:58 router tnp_sfm_3 mpc106 error address: 0x0a000000
Oct 26 12:05:58 router tnp_sfm_3 mpc106 PCI bus error status register: 0x02
Oct 26 12:05:58 router tnp_sfm_3 mpc106 was the PCI master
Oct 26 12:05:58 router tnp_sfm_3 C/BE bits: I/O read [0b0010]
Oct 26 12:05:58 router tnp_sfm_3 mpc106 error detection reg1: PCI cycle

```

```

Oct 26 12:05:58 router tnp_sfm_3 mpc106 PCI status reg: parity error
Oct 26 12:05:58 router tnp_sfm_3 ^B
Oct 26 12:05:58 router tnp_sfm_3 last message repeated 7 times
Oct 26 12:05:58 router tnp_sfm_3 Registers:
Oct 26 12:05:58 router tnp_sfm_3 R00: 0x06f5f81c R01: 0x06f5f9cc
R02: 0x00003344 R0
3: 0x00000000
Oct 26 12:05:58 router tnp_sfm_3 R04: 0x00008000 R05: 0x00000000
R06: 0x0010052c R0
7: 0x06f637e4
Oct 26 12:05:58 router tnp_sfm_3 R08: 0x06f5f81c R09: 0x00169810
R10: 0x000003b4 R1
1: 0x00000001
Oct 26 12:05:58 router tnp_sfm_3 R12: 0x00017b97 R13: 0xffffffff
R14: 0xffffffff R1
5: 0xffffffff
Oct 26 12:05:58 router tnp_sfm_3 R16: 0xffffffff R17: 0xffffffff
R18: 0xffffffff R1
9: 0xffffffff
Oct 26 12:05:58 router tnp_sfm_3 R20: 0xffffffff R21: 0xffffffff
R22: 0xffffffff R2
3: 0xffffffff
Oct 26 12:05:58 router tnp_sfm_3 R24: 0x00000003 R25: 0x00000000
R26: 0x00000001 R2
7: 0x0000fc78
Oct 26 12:05:58 router tnp_sfm_3 R28: 0x00150000 R29: 0x0016c4b0
R30: 0x06f5eb7c R3
1: 0x97c9c35e
Oct 26 12:05:58 router tnp_sfm_3 MSR: 0x0008b030 CTR: 0x000ac008
Link:0x06f5f81c SP
: 0x06f5f9cc
Oct 26 12:05:58 router tnp_sfm_3 CCR: 0x22200024 XER: 0x20000000
PC: 0x06f5f81c
Oct 26 12:05:58 router tnp_sfm_3 DSISR: 0x00000000 DAR: 0xffffffff
K_MSR: 0x0000103
0

```

Sample Output 2 The following sample output is another example of displaying the crash stack traceback and registration information:

```

root@host% vty fpc1

FPC160 platform (PPC 603e processor, 32Mb memory, 512Kb flash)

FPC1(host vty)# show nvram
System NVRAM :
  4080 available bytes, 4080 used, 0 free
  Contents:
0000000 R06: 0x0000005c R07: 0x850400d0
R08: 0x00000000 R09: 0x00000020 R10: 0x00000000 R11: 0x00000129
R12: 0x00000000 R13: 0x00000000 R14: 0x4005009a R15: 0x20000260
R16: 0xc8828784 R17: 0x84212800 R18: 0xc0004c61 R19: 0x80005900
R20: 0x80206000 R21: 0x84000304 R22: 0xd0410180 R23: 0x8c2005ac
R24: 0x00000003 R25: 0x00000000 R26: 0x00000001 R27: 0x0000fc48
R28: 0x001d0000 R29: 0x00000001 R30: 0x00136bb8 R31: 0x00000000
MSR: 0x0000b030 CTR: 0x001331e0 Link:0x000308c8 SP: 0x01baba34
CCR: 0x42200020 XER: 0x00000000 PC: 0x000308cc
DSISR: 0x00000000 DAR: 0xffffffff K_MSR: 0x00001030

```

Stack Traceback:

```

Frame 01: sp = 0x01baba34, pc = 0x000308c8
Frame 02: sp = 0x01babac4, pc = 0x0002647c
Frame 03: sp = 0x01babad4, pc = 0x00026590
Frame 04: sp = 0x01babadc, pc = 0x00106fcc
Frame 05: sp = 0x01babafc, pc = 0x00026620

```

ROM NVRAM:

```
0 available bytes, 0 used, 0 free
```

FPC1(host vty)# show syslog messages

```

[0+00:00:00.780 LOG: Info] Version 4.0R5 by tlim on 2000-08-10 04:45:54 UTC
[0+00:00:00.780 LOG: Info] On-board NVRAM contains diagnostic information.
[0+00:00:03.175 LOG: Info] PFEMAN: Established connection to Master
[Jan 30 21:53:05.804 LOG: Info] SNTPD: Initial time of day set.

```

What It Means Sample output 1 and 2 show the stack trace from the microkernel crash. Save the output from the `show nvram` and `show syslog` commands so that you can send them to JTAC when you open a case.

Step 2: Clear the NVRAM Contents

Purpose Currently the storage area for the logs on the NVRAM is limited to 4 KB. You need to delete old NVRAM logs to make room for new ones.

Action To clear the content of the NVRAM after you have captured the necessary information, follow these steps:

1. Exit from the CLI environment and create a UNIX-level shell by entering the `start shell` command:

```
user@host> start shell
```

2. Type `su` and the root password when prompted. You are now in the shell and the prompt is `%` instead of `>`, for example:

```

% su
Password: ****

```

3. Establish a vty session to the appropriate component. Use the `vtty` command followed by the abbreviation for the component, for example:

```
root@host% vtty sfm0
```

The vty prompt will vary depending on the component, for example:

```

SFM3(host vty)#
FPC1(host vty)#

```

4. Type the `clear nvram` command, for example:

```

SFM3(host vty)# clear nvram
FPC1(host vty)# clear nvram

```


Step 3: Check Packet Forwarding Engine Microkernel Core Files

Purpose If you observe disruption to the Packet Forwarding Engine microkernel, check the `/var/crash` directory for any core files created around the time of the crash.

Action To check the `/var/crash` directory, use the following JUNOS CLI operational mode command:

```
user@host> file list detail /var/crash
```

Sample Output

```
user@host> file list detail /var/crash
var/crash:
total 456630
-rw-r--r--  1 root  wheel   6814720 Dec 18 08:03 core-FPC4.100111808032
-rw-r--r--  1 root  wheel   65613824 Dec 10 04:58 core-SCB.100111004570
-rw-r--r--  1 root  wheel   65613824 Dec 19 00:23 core-SCB.100111900221
-rw-r--r--  1 root  wheel   65545216 Feb  9 20:46 core-SCB.101010920452
```

What It Means The sample output lists the contents of the `/var/crash/` directory. Check the date and timestamp for any core files created around the time of the crash. In the example above, four core files are listed.

Step 4: Collect and Send Crash Data to JTAC

Purpose If a Packet Forwarding Engine crash occurs on your router, collect the following data for JTAC evaluation and instruction.

Steps To Take To collect and send crash data to JTAC, follow these steps:

1. List the Core Files Generated by the Crash on page 315
2. Compress the Core Files on page 316
3. Log Software Version Information on page 316
4. Open a Case with JTAC on page 317

List the Core Files Generated by the Crash

Action To list the core files, follow these steps:

1. Exit from the CLI environment and create a UNIX-level shell by entering the `start shell` command:

```
user@host> start shell
```

2. Type `su` and the root password when prompted. You are now in the shell and the prompt is `%` instead of `>`, for example:

```
% su
Password: ****
```

3. Change the directory to `/var/crash` and type `ls -l`, for example:

```
root@host% cd /var/crash
root@host% ls -l
```

4. Look for any core files created around the time of the crash.

Sample Output

```
user@host> start shell
% su
Password: ****
root@host% cd /var/crash
root@host% ls -l
total 456630
-rw-r--r--  1 root  wheel   6814720 Dec 18 08:03 core-FPC4.100111808032
-rw-r--r--  1 root  wheel   65613824 Dec 10 04:58 core-SCB.100111004570
-rw-r--r--  1 root  wheel   65613824 Dec 19 00:23 core-SCB.100111900221
-rw-r--r--  1 root  wheel   65545216 Feb  9 20:46 core-SCB.101010920452
```

What It Means The sample output shows the current core files for the different components on the router; for example, `core-FPC4.100111808032` and `core-SCB.100111004570`.

Compress the Core Files

Purpose `gzip` is a compression utility used to compress the core files. Files created using the `gzip` command end with the file extension `.gz`. Compress the core files if they are larger than 50 MB.

Action To compress the core files with `gzip`, use the following command from the shell:

```
root@host% gzip filename
```

Sample Output

```
root@host% gzip core-SCB.101010920452
```

What It Means The contents of the core file are compressed into a single compressed file named `core-SCB.10101092045.gz`. The `gzip` command preserves the mode, ownership, and timestamps of files when compressing or decompressing them.

Log Software Version Information

Action To log JUNOS software version information, use the following JUNOS CLI operational mode command:

```
user@host> show version
```

Sample Output

```
user@host> show version
Hostname: host
Model: m10
JUNOS Base OS boot [5.0R5]
JUNOS Base OS Software Suite [5.0R5]
JUNOS Kernel Software Suite [5.0R5]
JUNOS Routing Software Suite [5.0R5]
JUNOS Packet Forwarding Engine Support [5.0R5]
JUNOS Crypto Software Suite [5.0R5]
JUNOS Online Documentation [5.0R5]
KERNEL 5.0R5 #0 built by builder on 2002-03-02 05:10:28 UTC
MGD release 5.0R5 built by builder on 2002-03-02 04:45:32 UTC
CLI release 5.0R5 built by builder on 2002-03-02 04:44:22 UTC
```

```

CHASSISD release 5.0R5 built by builder on 2002-03-02 04:43:37 UTC
DCD release 5.0R5 built by builder on 2002-03-02 04:42:47 UTC
RPD release 5.0R5 built by builder on 2002-03-02 04:46:17 UTC
SNMPD release 5.0R5 built by builder on 2002-03-02 04:52:26 UTC
MIB2D release 5.0R5 built by builder on 2002-03-02 04:45:37 UTC
APSD release 5.0R5 built by builder on 2002-03-02 04:43:31 UTC
VRRPD release 5.0R5 built by builder on 2002-03-02 04:52:34 UTC
ALARMD release 5.0R5 built by builder on 2002-03-02 04:43:24 UTC
PFED release 5.0R5 built by builder on 2002-03-02 04:46:06 UTC
CRAFTD release 5.0R5 built by builder on 2002-03-02 04:44:30 UTC
SAMPLED release 5.0R5 built by builder on 2002-03-02 04:52:20 UTC
ILMID release 5.0R5 built by builder on 2002-03-02 04:45:21 UTC
BPRELAYD release 5.0R5 built by builder on 2002-03-02 04:42:41 UTC
RMOPD release 5.0R5 built by builder on 2002-03-02 04:46:11 UTC
jkernel-dd release 5.0R5 built by builder on 2002-03-02 04:41:07 UTC
jroute-dd release 5.0R5 built by builder on 2002-03-02 04:41:21 UTC
jdocs-dd release 5.0R5 built by builder on 2002-03-02 04:39:11 UTC

```

What It Means The sample output shows the hostname, router model, and the different JUNOS software packages, processes, and documents.

Open a Case with JTAC

Action To open a case with JTAC, follow these steps:

1. Send an e-mail to support@juniper.net, and include the following information:
 - Output from the `show nvram` command
 - Output from the `show syslog messages` command
 - Output from the `show version` command
2. At the support engineer's request, ftp the `core-component.number.gz` file to a directory at [ftp.juniper.net](ftp://ftp.juniper.net). To ftp the core file to a directory, follow these steps:
 - a. At the shell prompt, enter `ftp ftp.juniper.net`.
 - b. At the name prompt, enter `anonymous`.
 - c. At the password prompt, enter your e-mail address as the password.
 - d. At the ftp prompt, enter the `cd pub/ incoming` string.
 - e. Enter the `mkdir case-number` command, where the `case-number` is the value of the case you opened with JTAC, for example, `1999-1231-9999`. If a directory has already been created, continue with the next step.
 - f. Enter the `cd case-number` command.
 - g. Enter the `binary` command so that the file transfer is in binary and not ASCII.
 - h. Enter the `put core-component.number.gz` command.

Sample Output The following output is an example of copying a core file from the shell to an ftp directory at ftp.juniper.net:

```
root@host% ftp ftp.juniper.net
Connected to colo-ftp.juniper.net.
220 colo-ftp.juniper.net FTP server (Version 6.00LS) ready.
Name (ftp.juniper.net: root): anonymous
331 Guest login ok, send your email address as password.
Password: ****
230 Guest login ok, access restrictions apply.
ftp> cd pub/ incoming
250 CWD command successful
ftp> mkdir 1999-1231-9999
257 MKD command successful.
ftp> cd 1999-1231-9999
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> put core-fpc4.100111808032.gz
```

What It Means The sample output shows that there is a connection to ftp.juniper.net, that the login name and password were entered, and that the core file was successfully copied from the shell to an ftp directory at ftp.juniper.net.