Monitoring System Processes and Logs

This chapter provides details on monitoring the health of the switch. It includes the following sections:

Displaying System Processes

Use the **show processes** command to obtain general information about all processes (see Examples 27-1 to 27-6).

Example 27-1 Displays System Processes

```
switch# show processes
PID    State  PC        Start_cnt    TTY   Process
-----  -----  --------  -----------  ----  -------------
  868      S  2ae4f33e            1     -  snmpd
  869      S  2acee33e            1     -  rscn
  870      S  2ac36c24            1     -  qos
  871      S  2ac44c24            1     -  port-channel
  872      S  2ac7a33e            1     -  ntp
    -     ER         -            1     -  mdog
    -     NR         -            0     -  vbuilder
```

Terms:

- PID = process ID.
- State = process state
  - D = uninterruptible sleep (usually IO)
  - R = runnable (on run queue)
  - S = sleeping
  - T = traced or stopped
  - Z = defunct (“zombie”) process
- NR = not-running
- ER = should be running but currently not-running
- PC = current program counter in hex format
- Start_cnt = how many times a process has been started (or restarted).
- TTY = terminal that controls the process. A “-” usually means a daemon not running on any particular TTY
- Process = name of the process

Example 27-2 Displays CPU Utilization Information

```
switch# show processes cpu
PID    Runtime(ms)  Invoked   uSecs  1Sec   Process
-----  -----------  --------  -----  -----  -----------
  842         3807    137001     27    0.0  sysmgr
 1112         1220     67974     17    0.0  syslogd
 1269          220     13568     16    0.0  fcfwd
 1276         2901     15419    188    0.0  zone
 1277          738     21010     35    0.0  xbar_client
 1278         1159      6789    170    0.0  wwn
 1279          515     67617      7    0.0  vsan
```

Terms:

- Runtime(ms) = CPU time the process has used, expressed in milliseconds
- Invoked = number of times the process has been invoked

- uSecs = microseconds of CPU time in average for each process invocation
- 1Sec = CPU utilization in percentage for the last one second

Example 27-3 Displays Process Log Information

```
switch# show processes log
Process           PID     Normal-exit  Stack-trace  Core     Log-create-time
----------------  ------  -----------  -----------  -------  ---------------
fspf              1339              N            Y        N  Jan  5 04:25
lcm               1559              N            Y        N  Jan  2 04:49
rib               1741              N            Y        N  Jan  1 06:05
```

Terms:

- Normal-exit = whether or not the process exited normally
- Stack-trace = whether or not there is a stack trace in the log
- Core = whether or not there exists a core file
- Log-create-time = when the log file got generated

Example 27-4 Displays Detail Log Information About a Process

```
switch# show processes log pid 1339
Service: fspf
Description: FSPF Routing Protocol Application

Started at Sat Jan  5 03:23:44 1980 (545631 us)
Stopped at Sat Jan  5 04:25:57 1980 (819598 us)
Uptime: 1 hours 2 minutes 2 seconds

Start type: SRV_OPTION_RESTART_STATELESS (23)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL (2)
Exit code: signal 9 (no core)
CWD: /var/sysmgr/work

Virtual Memory:

    CODE      08048000 - 0809A100
    DATA      0809B100 - 0809B65C
    BRK       0809D988 - 080CD000
    STACK     7FFFFD20
    TOTAL     23764 KB

Register Set:

    EBX 00000005         ECX 7FFFF8CC         EDX 00000000
    ESI 00000000         EDI 7FFFF6CC         EBP 7FFFF95C
    EAX FFFFFDFE         XDS 8010002B         XES 0000002B
    EAX 0000008E (orig)  EIP 2ACE133E         XCS 00000023
    EFL 00000207         ESP 7FFFF654         XSS 0000002B

Stack: 1740 bytes. ESP 7FFFF654, TOP 7FFFFD20

0x7FFFF654: 00000000 00000008 00000003 08051E95 ................
0x7FFFF664: 00000005 7FFFF8CC 00000000 00000000 ................
0x7FFFF674: 7FFFF6CC 00000001 7FFFF95C 080522CD ........\...."..
0x7FFFF684: 7FFFF9A4 00000008 7FFFFC34 2AC1F18C ........4......*
```

Example 27-5   Displays All Process Log Details

```
switch# show processes log details
======================================================
Service: snmpd
Description: SNMP Agent

Started at Wed Jan  9 00:14:55 1980 (597263 us)
Stopped at Fri Jan 11 10:08:36 1980 (649860 us)
Uptime: 2 days 9 hours 53 minutes 53 seconds

Start type: SRV_OPTION_RESTART_STATEFUL (24)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL (2)
Exit code: signal 6 (core dumped)
CWD: /var/sysmgr/work

Virtual Memory:

    CODE      08048000 - 0804C4A0
    DATA      0804D4A0 - 0804D770
    BRK       0804DFC4 - 0818F000
    STACK     7FFFFCE0
    TOTAL     26656 KB
..........
```

Example 27-6   Displays Memory Information About Processes

```
switch# show processes memory
PID    MemAlloc  StackBase/Ptr      Process
-----  --------  -----------------  ----------------
 1277    120632  7ffffcd0/7fffefe4  xbar_client
 1278     56800  7ffffce0/7ffffb5c  wwn
 1279   1210220  7ffffce0/7ffffbac  vsan
 1293    386144  7ffffcf0/7fffebd4  span
 1294   1396892  7ffffce0/7fffdff4  snmpd
 1295    214528  7ffffcf0/7ffff904  rscn
 1296     42064  7ffffce0/7ffffb5c  qos
```

Where:

- MemAlloc = total memory allocated by the process.
- StackBase/Ptr = process stack base and current stack pointer in hex format

Displaying System Status

Use the **show system** command to display system-related status information (Example 27-7 to Example 27-10.

Example 27-7 Displays Default Switch Port States

```
switch# show system default switchport
System default port state is down
System default trunk mode is on
```

Example 27-8 Displays Error Information for a Specified ID

```
switch# show system error-id 0x401D0019
Error Facility: module
Error Description: Failed to stop Linecard Async Notifciation.
```

Example 27-9 Displays the System Reset Information

```
switch# Show system reset-reason
----- reset reason for module 6 -----
1) At 520267 usecs after Tue Aug  5 16:06:24 1980
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 1.2(0.73a)
2) At 653268 usecs after Tue Aug  5 15:35:24 1980
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 1.2(0.45c)
3) No time
    Reason: Unknown
    Service:
    Version: 1.2(0.45c)
4) At 415855 usecs after Sat Aug  2 22:42:43 1980
    Reason: Power down triggered due to major temperature alarm
    Service:
    Version: 1.2(0.45c)
```

The **show system reset-reason** command displays the following information:

- In a Cisco MDS 9500 Series switch, the last four reset-reason codes for the supervisor module in slot #5 and slot #6 are displayed. If either supervisor module is absent, the reset-reason codes for that supervisor module are not displayed.
- In a Cisco MDS 9200 Series switch, the last four reset-reason codes for supervisor module in slot #1 are displayed.
- The **show system reset-reason module** *number* command displays the last four reset-reason codes for a specific module in a given slot. If a module is absent, then the reset-reason codes for that module will not be displayed.

Example 27-10 Displays System Uptime

```
switch# show system uptime
Start Time: Sun Oct 13 18:09:23 2030
Up Time:    0 days, 9 hours, 46 minutes, 26 seconds
```

Use the **show system resources** command to display system-related CPU and memory statistics (see Example 27-11).

Example 27-11 Displays System-Related CPU and Memory Information

```
switch# show system resources
Load average:   1 minute: 0.43   5 minutes: 0.17   15 minutes: 0.11
Processes   :   100 total, 2 running
CPU states  :   0.0% user,   0.0% kernel,   100.0% idle
Memory usage:   1027628K total,    313424K used,    714204K free
                   3620K buffers,   22278K cache
```

Where:

- Load is defined as number of running processes. The average reflects the system load over the past 1, 5, and 15 minutes.
- Processes displays the number of processes in the system, and how many are actually running when the command is issued.
- CPU states shows the CPU usage percentage in user mode, kernel mode, and idle time in the last one second.
- Memory usage provides the total memory, used memory, free memory, memory used for buffers, and memory used for cache in KB. Buffers and cache are also included in the *used* memory statistics.

Configuring Core and Log Files

You can save cores (from the active supervisor module, the standby supervisor module, or any switching module) to an external flash (slot 0) or to a TFTP server in one of two ways:

- On demand—to copy a single file based on the provided process ID.
- Periodically—to copy core files periodically as configured by the user.

To copy the core and log files on demand, follow this step:

	Command	Purpose
Step 1	`switch#` **`copy core:7407 slot0:coreSample`**	Copies the core file with the process ID 7407 as coreSample in slot 0.
	`switch#` **`copy core://5/1524 tftp:/1.1.1.1/abcd`**	Copies cores (if any) of a process with pid 1524 generated on slot 5 to tftp server.

- If the core file for the specified process ID is not available, you will see the following response:

  ```
  switch# copy core:133 slot0:foo
  No core file found with pid 133
  ```

- If two core files exist with same process ID, only one file will be copied:

  ```
  switch# copy core:7407 slot0:foo1
  2 core files found with pid 7407
  Only "/isan/tmp/logs/calc_server_log.7407.tar.gz" will be copied to the destination.
  ```

To copy the core and log files periodically, follow these steps:

	Command	Purpose
Step 1	`switch#` **`config t`**	Enters configuration mode.

	Command	Purpose
Step 2	`switch(config)# system cores slot0:coreSample`	Copies the core files coreSample to slot 0.
	`switch(config)# system cores tftp:/1.1.1.1/abcd`	Copies the core files (abcd) in the specified directory on the TFTP server.
	`switch(config)# no system cores`	Disable the core files copying feature.

A new scheme overwrites any previously-issued scheme. For example, if you issue a new system core command, the cores are periodically saved to the new location or file.

Tip Be sure to create any required directory before issuing this command. If the directory specified by this command does not exist, the switch software logs a syslog message each time a copy cores is attempted.)

Clearing the Core Directory

Use the **clear cores** command to clean out the core directory. The software keeps the last few cores per service and per slot and clears all other cores present on the active supervisor module.

```
switch# clear cores
```

Displaying Cores Status

Use the **show system cores** command to display the currently configured scheme for copying cores. See Examples 27-12 to 27-14.

Example 27-12 Displays the status of System Cores

```
switch# show system cores
Transfer of cores is enabled
```

Example 27-13 Displays All Cores Available for Upload from the Active Supervisor Module

```
switch# show cores
Module-num Process-name PID Core-create-time
---------- ------------ --- ----------------
5           fspf        1524    Jan 9 03:11
6           fcc         919     Jan 9 03:09
8           acltcam     285     Jan 9 03:09
8           fib         283     Jan 9 03:08
```

Where:

`module-num` shows the slot number on which the core was generated. In this example, the `fspf core` was generated on the active supervisor module (slot 5), `fcc` was generated on the standby supervisor module (slot 6), and `acltcam` and `fib` were generated on the switching module (slot 8).

Example 27-14 Displays Logs on the Local System

```
switch# show processes log
Process PID     Normal-exitStack-traceCore Log-create-time
---------------- ------ ----------- ----------- -------- ---------------
fspf    1524    N          Y          Y   Jan 9 03:11
```

Configuring HA Policy

You can disable the HA policy supervisor reset feature (enabled by default) for debugging and troubleshooting purposes.

To configure HA policies, follow this step:

	Command	Purpose
Step 1	switch# **system no hap-reset**	Disables supervisor reset HA policy.
	switch# **system hap-reset**	Enables Supervisor Reset HA policy whenever a critical service runs out of HA policies (default) and reverts it to factory default.

Resetting HA Statistics

The system statistics reset feature resets the high availability statistics collected by the system.

```
switch# system statistics reset
```

Configuring Heartbeat Checks

The software monitors every service to verify if heartbeats are sent at regular intervals. If not, the software restarts that service. This feature helps locate situations when a service is stuck in an infinite loop.

You can disable the heartbeat checking feature (enabled by default) for debugging and troubleshooting purposes like attaching a GDB to a specified process.

To configure heartbeat checks, follow this step:

	Command	Purpose
Step 1	switch# **system no heartbeat**	Disables heartbeat checks.
	switch# **system heartbeat**	Enables heartbeat checks (default) and reverts it to factory default.

Configuring Watchdog Checks

If a watchdog is not logged at every 8 seconds by the software, the supervisor module reboots the switch.

You can disable the watchdog checking feature (enabled by default) for debugging and troubleshooting purposes like attaching a GDB or a kernel GDB (KGDB) to a specified process.

To configure watchdog checks, follow this step:

	Command	Purpose
Step 1	switch# **system no watchdog**	Disables watchdog checks.
	switch# **system watchdog**	Enables watchdog checks (default) and reverts it to factory default.

Configuring Upgrade Resets

This feature enables supervisor module resets when an upgrade has failed. If the upgrade fails for any reason, the software reboots the switch since the file system may be in an unstable state.

You can disable the upgrade-reset feature (enabled by default) for debugging and troubleshooting purposes.

To configure supervisor upgrade resets, follow this step:

	Command	Purpose
Step 1	`switch# system no upgrade-reset`	Disables the upgrade reset feature.
	`switch# system upgrade-reset`	Enables the upgrade reset feature (default) and reverts it to factory default.

Configuring Kernel Core Dumps

Caution Changes to the kernel cores should be made by an administrator or individual who is completely familiar with switch operations.

When a specific module's operating system (OS) crashes, it is sometimes useful to obtain a full copy of the memory image (called a kernel core dump) to identify the cause of the crash. When the module experiences a kernel core dump it triggers the proxy server configured on the supervisor. The supervisor sends the module's OS kernel core dump to the Cisco MDS 9000 System Debug Server. Similarly, if the supervisor OS fails the supervisor sends its OS kernel core dump to the Cisco MDS 9000 System Debug Server.

Note The Cisco MDS 9000 System Debug Server is a Cisco application that runs on Linux. It creates a repository for kernel core dumps. You can download the Cisco MDS 9000 System Debug Server from the Cisco.com website at http://www.cisco.com/kobayashi/sw-center/sw-stornet.shtml.

Kernel core dumps are only useful to your technical support representative. The kernel core dump file, which is a large binary file, must be transferred to an external server that resides on the same physical LAN as the switch. The core dump is subsequently interpreted by technical personnel who have access to source code and detailed memory maps.

Tip Core dumps take up disk space on the Cisco MDS 9000 System Debug Server application. If all levels of core dumps (**level all** option) are configured, you need to ensure that a minimum of 1GB of disk space is available on the Linux server running the Cisco MDS 9000 System Debug Server application to accept the dump. If the process does not have sufficient space to complete the generation, the module resets itself.

To configure the external server, follow these steps:

	Command	Purpose

Step 1	`switch# config terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)# kernel core target 10.50.5.5`
	`succeeded`
		Configures the external server's IP address.

To configure the module information, follow these steps:

	Command	Purpose

Step 1	`switch# config terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	`switch(config)# kernel core module 5`
	`succeeded`
		Configures kernel core generation for module 5.

	`switch(config)# kernel core module 5 level header`
	`succeeded`
		Configures kernel core generation for module 5, and limits the generation to header-level cores.

Step 3	`switch(config)# kernel core limit 2`
	`succeeded`
		Configures generations for two modules. The default is 1 module.

All changes made to kernel cores are saved to the running configuration and may be viewed using the **show running-config** command. Alternatively, use the **show kernel cores** command to view specific configuration changes (see examples 27-15 to 27-17).

Example 27-15 Displays the Core Limit

```
switch# show kernel core limit
2
```

Example 27-16 Displays the External Server

```
switch# show kernel core target
10.50.5.5
```

Example 27-17 Displays the Core Settings for the Specified Module

```
switch# show kernel core module 5
module 5 core is enabled
         level is header
         dst_ip is 10.50.5.5
         src_port is 6671
         dst_port is 6666
         dump_dev_name is eth1
         dst_mac_addr is 00:00:0C:07:AC:01
```