Configuring Onboard Failure Logging

This chapter contains the following sections:

About OBFL

Cisco NX-OS provides the ability to log failure data to persistent storage, which you can retrieve and display for analysis at a later time. This onboard failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. The information will help analyze failed modules.

OBFL stores the following types of data:

- Time of initial power-on

- Slot number of the module in the chassis

- Initial temperature of the module

- Firmware, BIOS, FPGA, and ASIC versions

- Serial number of the module

- Stack trace for crashes

- CPU hog information

- Memory leak information

- Software error messages

- Hardware exception logs

- Environmental history

- OBFL-specific history information

- ASIC interrupt and error statistics history

- ASIC register dumps

Prerequisites for OBFL

You must have network-admin user privileges.

Guidelines and Limitations for OBFL

OBFL has the following guidelines and limitations:

- OBFL is enabled by default.

- OBFL flash supports a limited number of writes and erases. The more logging you enable, the faster you use up this number of writes and erases.

- The **show system reset-reason module** *module num* command does not display the reset reason incase of a module failure. Due to lack of persistent storage of the module reset-reason, this command is not effective after a reboot. Since the exception log is available in persistent storage, after a reboot, you can you can view the reset-reason using the **show logging onboard exception-log** command.

Note   Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Default Settings for OBFL

The following table lists the default settings for OBFL parameters.

Parameters	Default
OBFL	All features enabled

Configuring OBFL

You can configure the OBFL features on Cisco NX-OS devices.

Before you begin

Make sure that you are in global configuration mode.

Procedure

	Command or Action	**Purpose**

Step 1	**configure terminal**
	
	Example:
	
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hw-module logging onboard**
	
	Example:
	
	`switch(config)# hw-module logging`
	`onboard`
	`Module: 7 Enabling ... was successful.`
	`Module: 10 Enabling ... was successful.`
	`Module: 12 Enabling ... was successful.`
		Enables all OBFL features.

Step 3	**hw-module logging onboard counter-stats**
	
	Example:
	
	`switch(config)# hw-module logging`
	`onboard counter-stats`
	`Module: 7 Enabling counter-stats ...`
	`was successful.`
	`Module: 10 Enabling counter-stats ...`
	`was successful.`
	`Module: 12 Enabling counter-stats ...`
	`was successful.`
		Enables the OBFL counter statistics.

Step 4	**hw-module logging onboard cpuhog**
	
	Example:
	
	`switch(config)# hw-module logging`
	`onboard cpuhog`
	`Module: 7 Enabling cpu-hog ... was`
	`successful.`
	`Module: 10 Enabling cpu-hog ... was`
	`successful.`
	`Module: 12 Enabling cpu-hog ... was`
	`successful.`
		Enables the OBFL CPU hog events.

Step 5	**hw-module logging onboard environmental-history**
	
	Example:
	
	`switch(config)# hw-module logging`
	`onboard environmental-history`
	`Module: 7 Enabling environmental-history`
	` ... was successful.`
	`Module: 10 Enabling`
	`environmental-history ... was`
	`successful.`
	`Module: 12 Enabling`
	`environmental-history ... was`
	`successful.`
		Enables the OBFL environmental history.

Step 6	**hw-module logging onboard error-stats**	Enables the OBFL error statistics.

	Command or Action	Purpose
	Example: ```switch(config)# hw-module logging onboard error-stats Module: 7 Enabling error-stats ... was successful. Module: 10 Enabling error-stats ... was successful. Module: 12 Enabling error-stats ... was successful.```	
Step 7	**hw-module logging onboard interrupt-stats** **Example:** ```switch(config)# hw-module logging onboard interrupt-stats Module: 7 Enabling interrupt-stats ... was successful. Module: 10 Enabling interrupt-stats ... was successful. Module: 12 Enabling interrupt-stats ... was successful.```	Enables the OBFL interrupt statistics.
Step 8	**hw-module logging onboard module** *slot* **Example:** ```switch(config)# hw-module logging onboard module 7 Module: 7 Enabling ... was successful.```	Enables the OBFL information for a module.
Step 9	**hw-module logging onboard obfl-logs** **Example:** ```switch(config)# hw-module logging onboard obfl-logs Module: 7 Enabling obfl-log ... was successful. Module: 10 Enabling obfl-log ... was successful. Module: 12 Enabling obfl-log ... was successful.```	Enables the boot uptime, device version, and OBFL history.
Step 10	(Optional) **show logging onboard** **Example:** ```switch(config)# show logging onboard```	Displays information about OBFL.
Step 11	(Optional) **copy running-config startup-config** **Example:** ```switch(config)# copy running-config startup-config```	Copies the running configuration to the startup configuration.

Verifying the OBFL Configuration

To display OBFL information stored in flash on a module, perform one of the following tasks:

Command	Purpose
show logging onboard boot-uptime	Displays the boot and uptime information.
show logging onboard counter-stats	Displays statistics on all ASIC counters.
show logging onboard credit-loss	Displays OBFL credit loss logs.
show logging onboard device-version	Displays device version information.
show logging onboard endtime	Displays OBFL logs to a specified end time.
show logging onboard environmental-history	Displays environmental history.
show logging onboard error-stats	Displays error statistics.
show logging onboard exception-log	Displays exception log information.
show logging onboard interrupt-stats	Displays interrupt statistics.
show logging onboard module *slot*	Displays OBFL information for a specific module.
show logging onboard obfl-history	Displays history information.
show logging onboard obfl-logs	Displays log information.
show logging onboard stack-trace	Displays kernel stack trace information.
show logging onboard starttime	Displays OBFL logs from a specified start time.
show logging onboard status	Displays OBFL status information.

Use the **show logging onboard status** command to display the configuration status of OBFL.

```
switch# show logging onboard status
--------------------------
OBFL Status
--------------------------
Switch OBFL Log: Enabled

Module: 4 OBFL Log: Enabled
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled

Module: 22 OBFL Log: Enabled
```

```
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled
```

Use the **clear logging onboard** command to clear the OBFL information for each of the **show** command options listed.

Configuration Example for OBFL

This example shows how to enable OBFL on module 2 for environmental information:

```
switch# configure terminal
switch(config)# hw-module logging onboard module 2 environmental-history
```

Additional References

Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 3600 NX-OS Fundamentals Configuration Guide*