

Onboard Failure Logging

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.

- Overview, on page 1
- Guidelines and Limitations, on page 2
- Configuration, on page 2
- Related Documents, on page 6

Overview

OBFL (On-Board Failure Logging) is a feature that stores critical operational and diagnostic data for hardware modules.

Types of Data Stored by OBFL

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

Default Settings

The following table lists the default settings for OBFL parameters.

Parameters	Default
OBFL	All features enabled

Guidelines and Limitations

Prerequisites

You must have network-admin user privileges.

General Guidelines

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 that you enable, the faster you use up this number of writes and erases.



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

Configuration

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

Before you begin

Make sure that you are in global configuration mode.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable all OBFL features using the command **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.
```

Step 3 Enable the OBFL counter statistics using the command **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.
```

Step 4 Enable the OBFL CPU hog events using the command **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.
```

Step 5 Enable the OBFL environmental history using the command **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.
```

Step 6 Enable the OBFL error statistics using the command **hw-module logging onboard error-stats**

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 Enable the OBFL interrupt statistics using the command **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.
```

Step 8 Enable the OBFL information for a module using the command **hw-module logging onboard module** *slot* and enable the boot uptime, device version, and OBFL history using the command **hw-module logging onboard obfl-logs**

Example:

```
switch(config)# hw-module logging onboard module 7
 Module: 7 Enabling ... was successful.
```

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.
```

Step 9 (Optional) Display information about OBFL using the command **show logging onboard** and copy the running configuration to the startup configuration using the command **copy running-config startup-config** .

Example:

```
switch(config)# show logging onboard
```

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Note To display OBFL information stored in flash on a module, see Verifying the OBFL Configuration .

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
```

What to do next

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.

Command	Purpose
show logging onboard module *slot* **internal reset-reason**	Displays OBFL information for a specific module. **Note** If you specify **internal reset-reason** and you are operating in a redundant supervisor configuration, checking the persistent log on the standby supervisor after a system reset occurs will display a relevant reset reason. The reset reason is recorded on the on-board flash for both the active and standby supervisor.
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.

Related Documents

This reference provides a related document for configuration files.

Table 1: Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*