



Troubleshooting

This chapter describes how to identify and resolve software problems related to the Cisco IOS software on the CGR 2010 ESM.

You can use the command-line interface (CLI) to identify and solve problems.



Note

For complete syntax and usage information for the commands used in this chapter, see the command reference for this release and the *Cisco IOS Command Summary, Release 12.2*.

- [Recovering from a Software Failure, page 18-2](#)
- [Preventing Autonegotiation Mismatches, page 18-3](#)
- [Preventing Autonegotiation Mismatches, page 18-3](#)
- [Troubleshooting Power over Ethernet Switch Module Ports, page 18-3](#)
- [SFP Module Security and Identification, page 18-4](#)
- [Monitoring Temperature and Configuring the Yellow Threshold, page 18-5](#)
- [Using Ping, page 18-6](#)
- [Using Layer 2 Traceroute, page 18-9](#)
- [Using IP Traceroute, page 18-11](#)
- [Using TDR, page 18-12](#)
- [Using Debug Commands, page 18-13](#)
- [Using the show platform forward Command, page 18-15](#)
- [Using the crashinfo File, page 18-17](#)

Recovering from a Software Failure

The switch module software can be corrupted during an upgrade by downloading the wrong file to the switch module, and by deleting the image file. In all of these cases, the switch module does not pass the power-on self-test (POST), and there is no connectivity.

Recovery Procedure at 115200 Baud Line Speed

This procedure uses the Xmodem Protocol at 115200 baud line speed to recover from a corrupt or wrong image file. There are many software packages that support the Xmodem Protocol, and this procedure is largely dependent on the emulation software that you are using.

Because there is no console port on the switch module, recovery must be done from the CGR 2010 router.

For more information, see “[Recovering from a Corrupted Software Image Using Xmodem](#)” in the *Cisco Enhanced EtherSwitch Services Modules Configuration Guide*.

Recovering from a Lost or Forgotten Password

If you lose or forget your password, you can delete the switch module password and set a new one.

Before you begin, make sure that:

- You have physical access to the switch module.
- At least one switch module port is enabled and it is not connected to a device.

To delete the switch module password and set a new one, follow these steps:

-
- Step 1** Press the **Express Setup** button until the SETUP LED blinks green and the LED of an available downlink port blinks green.
- a. If no switch module downlink port is available for your PC or laptop connection, disconnect a device from one of the other downlink ports.
 - b. Press the **Express Setup** button again until the SETUP LED and the port LED blink green.
- Step 2** Connect your PC or laptop to the port with the blinking green LED.
The SETUP LED and the switch module downlink port LED stop blinking and stay solid green.
- Step 3** Press and hold the **Express Setup** button.
Notice that the SETUP LED starts blinking green again.
- a. Continue holding the button until the SETUP LED turns solid green (approximately 5 seconds).
Release the Express Setup button immediately.
- This procedure deletes the password without affecting any other configuration settings. You can now access the switch module without a password through the console port or by using the device manager.
- Step 4** Enter a new password through the device manager by using the Express Setup window or through the command line interface by using the **enable secret** global configuration command.
-

Preventing Autonegotiation Mismatches

The IEEE 802.3ab autonegotiation protocol manages the switch module settings for speed (10, 100, and 1000 Mbps, excluding SFP module ports) and duplex (half or full). There are situations when this protocol can incorrectly align these settings, reducing performance. A mismatch occurs under these circumstances:

- A manually set speed or duplex parameter is different from the manually set speed or duplex parameter on the connected port.
- A port is set to autonegotiate, and the connected port is set to full duplex with no autonegotiation.

To maximize switch module performance and ensure a link, follow one of these guidelines when changing the settings for duplex and speed:

- Let both ports autonegotiate both speed and duplex.
- Manually set the speed and duplex parameters for the ports on both ends of the connection.

**Note**

If a remote device does not autonegotiate, configure the duplex settings on the two ports to match. The speed parameter can adjust itself even if the connected port does not autonegotiate.

Troubleshooting Power over Ethernet Switch Module Ports

These sections describe how to troubleshoot Power over Ethernet (PoE) ports.

Disabled Port Caused by Power Loss

If a powered device (such as a Cisco IP Phone 7910) that is connected to a PoE switch module port and is powered by an AC power source loses power from the AC power source, the device might enter an error-disabled state.

Recovering from an Error-Disabled State

1. Enter the **shutdown** interface configuration command.
2. Then enter the **no shutdown** interface command.

You can also configure automatic recovery on the switch module to recover from the error-disabled state. The **errdisable recovery cause loopback** and the **errdisable recovery interval seconds** global configuration commands automatically take the interface out of the error-disabled state after the specified period of time.

Use these commands, described in the command reference for this release, to monitor the PoE port status:

- **show controllers power inline** privileged EXEC command
- **show power inline** privileged EXEC command
- **debug ilpower** privileged EXEC command

Disabled Port Caused by False Link-Up

If a Cisco powered device is connected to a port and you configure the port by using the **power inline never interface** configuration command, a false link-up can occur, placing the port into an error-disabled state.

To take the port out of the error-disabled state, enter the **shutdown interface** and the **no shutdown interface** configuration commands.

Do not connect a Cisco powered device to a port that has been configured with the **power inline never** command.

SFP Module Security and Identification

Cisco small form-factor pluggable (SFP) modules have a serial EEPROM that contains the module serial number, the vendor name and ID, a unique security code, and cyclic redundancy check (CRC). When an SFP module is inserted in the switch module, the switch module software reads the EEPROM to verify the serial number, vendor name and vendor ID, and recompute the security code and CRC. If the serial number, the vendor name or vendor ID, the security code, or CRC is invalid, the software generates a security error message and places the interface in an error-disabled state.



Note

The security error message references the GBIC_SECURITY facility. The switch module supports SFP modules and does not support GBIC modules. Although the error message text refers to GBIC interfaces and modules, the security messages actually refer to the SFP modules and module interfaces. For more information about error messages, see the system message guide for this release.

Non-Cisco SFP Module

1. Remove the SFP module from the switch module, and replace it with a Cisco module.
2. After inserting a Cisco SFP module, use the **errdisable recovery cause gbic-invalid** global configuration command to verify the port status, and enter a time interval for recovering from the error-disabled state.

After the elapsed interval, the switch module brings the interface out of the error-disabled state and retries the operation.

For more information about the **errdisable recovery** command, see the command reference for this release.

Cisco SFP Module

If the module is identified as a Cisco SFP module, but the system is unable to read vendor-data information to verify its accuracy, an SFP module error message is generated.

In this case, you should remove and re-insert the SFP module. If it continues to fail, the SFP module might be defective.

Monitoring SFP Module Status

You can check the physical or operational status of an SFP module by using the **show interfaces transceiver** privileged EXEC command. This command shows the operational status, such as the temperature and the current for an SFP module on a specific interface and the alarm status.



Note

The **show interfaces transceiver** command only works if the SFP supports DOM.

You can also use the command to check the speed and the duplex settings on an SFP module. For more information, see the **show interfaces transceiver** command in the command reference for this release.

Monitoring Temperature and Configuring the Yellow Threshold

The CGR 2010 ESM includes sensors that measure and monitor the status and internal temperature of critical components. Internal component temperatures are measured for the central processor, internal components, and interface cards.

The measured temperature is compared to predetermined threshold limits and, if the temperature does not fall within the limits, the information is recorded and a warning sent to the system administrator by means of Simple Network Management Protocol (SNMP) traps until the temperature falls back to its normal range.

- Use the **show env temperature status** privileged EXEC command to display the current temperature value, state, and thresholds of the switch's CPU and Ethernet board (see [Table 18-1](#) below). The temperature value is the temperature in the switch module (not the external temperature). If the temperature exceeds the threshold, a warning message is sent.
- You can configure the yellow threshold level (in Celsius) by using the **system env temperature threshold yellow value** global configuration command to set the difference between the yellow and red thresholds. You cannot configure the green or red thresholds (see "[Configuring the Yellow Threshold](#)" below for details).

Temperature Show Commands

The switch module monitors the temperature conditions to determine the health of the power supplies. The temperature value is the temperature in the switch module (not the external temperature).

[Table 18-1](#) describes the **Show** commands that monitor the switch's temperature:

Table 18-1 Temperature Show Commands

| Command | Function |
|---|---|
| Switch# show env all <i>temperature status</i> | <ul style="list-style-type: none"> • <i>all</i>: Displays all environmental monitor parameters • <i>temperature status</i>: Shows temperature status and threshold levels of the switch's CPU and Ethernet board. |

Configuring the Yellow Threshold

You cannot configure the green and red thresholds but you can configure the yellow threshold.

**Note**

The yellow threshold is independent for the CPU sensor and Ethernet board sensor. At this time, only the CPU yellow threshold can be changed.

Use the **system env temperature threshold yellow *value*** global configuration command to specify the difference between the yellow and red threshold values and to configure the yellow threshold (in Celsius). The range is 20 to 25. The default value is 20.

For example, if the red threshold is 60 degrees C and you want to configure the yellow threshold as 51 degrees C, set the difference between the thresholds as 15 by using the **system env temperature threshold yellow 9** command.

Use the no form of this command to return to the default value.

**Note**

The default red threshold is 100 degrees C; the default yellow threshold is 94 degrees C for the CPU board; and 95 degrees C for the Ethernet board.

Using Ping

- [Understanding Ping, page 18-6](#)
- [Using Ping, page 18-6](#)

Understanding Ping

The switch module supports IP ping, which you can use to test connectivity to remote hosts. Ping sends an echo request packet to an address and waits for a reply.

The switch module also provides the Control Plane Security feature, which by default drops ping response packets received on user network interfaces (UNIs) or enhanced network interfaces (ENIs). However, methods are available to ping successfully from the switch module to a host connected to a UNI or ENI.

Control Plane Security does not drop ping response packets to or from network node interfaces (NNIs), and no special configuration is required to enable pings to or from hosts connected to NNIs.

Using Ping

Beginning in privileged EXEC mode, use the **ping** command to ping another device on the network from the switch module:

| Command | Description |
|--|---|
| ping [<i>host</i> <i>address</i>] | Ping a remote host by supplying the hostname or IP network address. Note Though other protocol keywords are available with the ping command, they are not supported in this release. |



Note

Ping is not supported on a UNI or ENI configured as an IEEE 802.1Q tunnel port. Ping is supported on NNIs on all software images.

It is important to note that the software images available for the switch module provide different options for pinging a host connected to a UNI or ENI.

The next sections apply to both access ports and trunk ports.

All Software Versions

For all software images for the switch module, you can use a Layer 3 service policy to enable pings from the switch module to a host connected to a UNI or ENI.



Note

For a switch module running the IP services image, IP routing is not enabled by default and does not have to be enabled to use a Layer 3 service policy.

This example is one possible configuration:

```
switch# configure terminal
switch(config)# access list 101 permit ip any any
switch(config)# class-map match-any ping-class
switch(config-cmap)# match access-group 101
switch(config-cmap)# exit
switch(config)# policy-map ping-policy
switch(config-pmap)# class ping-class
switch(config-pmap-c)# police 1000000
switch(config-pmap-c)# exit
switch(config-pmap)# exit
switch(config)# int fa0/1
switch(config-if)# service-policy input ping-policy
switch(config-if)# switchport access vlan 2
switch(config-if)# no shut
switch(config-if)# exit
switch(config)# int vlan 2
switch(config-if)# ip address 192.168.1.1 255.255.255.0
switch(config-if)# end
switch# ping 192.168.1.2
```

IP Services Image

When your switch module is running the IP services image, you can use any of these methods:

- Apply a Layer 3 service policy to a UNI or ENI.
- Enable IP routing globally and ping from a Switch Virtual Interface (SVI).
- Enable IP routing and ping from a routed port.

For a sample configuration of how to add a Layer 3 service policy to a UNI or ENI, see the [“All Software Versions” section on page 18-7](#).

For examples using IP routing and pinging from an SVI or a routed port, see the next sections.

IP Routing and SVI

IP routing is only supported when the switch module is running the IP services image.

You can use this configuration to enable IP routing and enable pings from an SVI to a host connected to a UNI or ENI.

```
Switch# configure terminal
Switch(config)# ip routing
Switch(config)# int fa0/1
Switch(config-if)# switchport access vlan 2
Switch(config-if)# no shutdown
Switch(config-if)# int vlan 2
Switch(config-if)# ip address 192.168.1.1 255.255.255.0
Switch(config-if)# end
Switch# ping 192.168.1.2
```

With this configuration, a host with an IP address of 192.168.1.2 can be pinged from the switch module.

IP Routing and Routed Port

You can use this configuration to enable IP routing, change a switchport to a routed port, and permit pings from the switch module to a connected host:

```
switch# configure terminal
switch(config)# int fa0/1
switch(config-if)# no switchport
switch(config-if)# ip address 192.168.1.1 255.255.255.0
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# ip routing
switch(config)# end
switch# ping 192.168.1.2
```

Ping Responses

This response is typical of a successful ping to a host:

```
Switch# ping 72.20.52.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 72.20.52.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Switch#
```

An unsuccessful ping results in this message:

```
Switch# ping 72.20.52.3
```



```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echoes to 172.20.52.3, timeout is 2 seconds:  
. . . . .  
Success rate is 0 percent (0/5)
```

Summary

Keep these guidelines in mind while pingging:

- IP routing is available only with the IP services image and is disabled by default.
- To ping a host in a different IP subnetwork from the switch module, you must have IP routing configured to route between the subnets, and a static route to the destination might also be appropriate.
- All software versions can use a Layer 3 service policy to permit pings to and from a host connected to a UNI or ENI. For more information about policy maps, see the [“Input and Output Policies” section on page 14-4](#).

If your switch module is running the IP services image, use one of these methods to ping a host connected to a UNI or ENI:

- Use a Layer 3 service policy to permit pings to and from a host connected to a UNI or ENI.
- Enable global IP routing and configure a port as a routed port by using the **no switchport** interface configuration command.
- Enable global IP routing, create an SVI, and assign an IP address to it

Using Layer 2 Traceroute

- [Understanding Layer 2 Traceroute, page 18-9](#)
- [Layer 2 Traceroute Usage Guidelines, page 18-10](#)
- [Displaying the Physical Path, page 18-10](#)

Understanding Layer 2 Traceroute

The Layer 2 traceroute feature allows the switch module to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. It finds the path by using the MAC address tables of the switches in the path. When the switch module detects a device in the path that does not support Layer 2 traceroute, the switch module continues to send Layer 2 trace queries and lets them time out.



Note

Layer 2 traceroute is available only on NNIs.

The switch module can only identify the path from the source device to the destination device. It cannot identify the path that a packet takes from source host to the source device or from the destination device to the destination host.

Layer 2 Traceroute Usage Guidelines

- Cisco Discovery Protocol (CDP) must be enabled on all the devices in the network. For Layer 2 traceroute to function properly, do not disable CDP.



Note CDP is enabled by default on NNIs. You can enable CDP on ENIs, but UNIs do not support CDP.

If any devices in the physical path are transparent to CDP, the switch module cannot identify the path through these devices.

- A switch module is reachable from another switch module when you can test connectivity by using the **ping** privileged EXEC command. All switches in the physical path must be reachable from each other.
- The maximum number of hops identified in the path is ten.
- You can enter the **traceroute mac** or the **traceroute mac ip** privileged EXEC command on a switch module that is not in the physical path from the source device to the destination device. All switches in the path must be reachable from this switch module.
- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.
- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.
- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.
- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the switch module uses the Address Resolution Protocol (ARP) to associate the IP addresses with the corresponding MAC addresses and the VLAN IDs.
 - If an ARP entry exists for the specified IP address, the switch module uses the associated MAC address and identifies the physical path.
 - If an ARP entry does not exist, the switch module sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.
- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute feature is not supported. When more than one CDP neighbor is detected on a port, the Layer 2 path is not identified, and an error message appears.
- This feature is not supported in Token Ring VLANs.

Displaying the Physical Path

You can display the physical path that a packet takes from a source device to a destination device by using one of these privileged EXEC commands:

- **tracetroute mac** [**interface** *interface-id*] {*source-mac-address*} [**interface** *interface-id*] {*destination-mac-address*} [**vlan** *vlan-id*] [**detail**]

- **tracetroute mac ip** {*source-ip-address* | *source-hostname*} {*destination-ip-address* | *destination-hostname*} [**detail**]

**Note**

Layer 2 traceroute is available only on NNIs.

For more information, see the command reference for this release.

Using IP Traceroute

- [Understanding IP Traceroute, page 18-11](#)
- [Executing IP Traceroute, page 18-11](#)

Understanding IP Traceroute

You can use IP traceroute to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

Your switches can participate as the source or destination of the **tracetroute** privileged EXEC command and might or might not appear as a hop in the **tracetroute** command output. If the switch module is the destination of the traceroute, it is displayed as the final destination in the output. Intermediate switches do not show up in the output if they are only bridging the packet from one port to another within the same VLAN. However, if the intermediate switch module is a multilayer switch module that is routing a particular packet, this switch module shows up as a hop in the output.

The **tracetroute** privileged EXEC command uses the Time To Live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends an Internet Control Message Protocol (ICMP) time-to-live-exceeded message to the sender. Traceroute finds the address of the first hop by examining the source address field of this message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-to-live-exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To learn when a datagram reaches its destination, traceroute sets the UDP destination port number in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram destined to itself containing a destination port number that is unused locally, it sends an ICMP *port-unreachable* error to the source. Because all errors except port-unreachable errors come from intermediate hops, the receipt of a port-unreachable error means that this message was sent by the destination port.

Executing IP Traceroute

Beginning in privileged EXEC mode, follow this step to trace that the path packets take through the network:

| Command | Description |
|---------------------------------|---|
| <code>traceroute ip host</code> | Trace the path that packets take through the network. |

**Note**

Though other protocol keywords are available with the **traceroute** privileged EXEC command, they are not supported in this release.

This example shows how to perform a **traceroute** to an IP host:

```
Switch# traceroute ip 171.9.15.10

Type escape sequence to abort.
Tracing the route to 171.69.115.10
 0 172.2.52.1 0 msec 0 msec 4 msec
 1 172.2.1.203 12 msec 8 msec 0 msec
 2 171.9.16.6 4 msec 0 msec 0 msec
 3 171.9.4.5 0 msec 4 msec 0 msec
 4 171.9.121.34 0 msec 4 msec 4 msec
 5 171.9.15.9 120 msec 132 msec 128 msec
 6 171.9.15.10 132 msec 128 msec 128 msec
Switch
```

The display shows the hop count, IP address of the router, and the round-trip time in milliseconds for each of the three probes that are sent.

Table 18-2 Traceroute Output Display Characters

| Character | Description |
|-----------|--|
| * | Probe timed out |
| ? | Unknown packet type |
| A | Administratively unreachable—usually this output means that an access list is blocking traffic |
| H | Host unreachable |
| N | Network unreachable |
| P | Protocol unreachable |
| Q | Source quench |
| U | Port unreachable |

Using TDR

- [Understanding TDR, page 18-13](#)
- [Running TDR and Displaying the Results, page 18-13](#)

Understanding TDR

You can use the Time Domain Reflector (TDR) feature to diagnose and resolve cabling problems. When running TDR, a local device sends a signal through a cable and compares the reflected signal to the initial signal.

On the CGR 2010 ESM, TDR is supported only on the copper Ethernet 10/100 ports or on dual-purpose ports configured to either 10 Mbps or 100 Mbps ports and media-type RJ-45.

TDR can detect these cabling problems:

- Open, broken, or cut twisted-pair wires—The wires are not connected to the wires from the remote device.
- Shorted twisted-pair wires—The wires are touching each other or the wires from the remote device. For example, a shorted twisted pair can occur if one wire of the twisted pair is soldered to the other wire.

If one of the twisted-pair wires is open, TDR can find the length at which the wire is open.

Use TDR to diagnose and resolve cabling problems in these situations:

- Replacing a switch module
- Setting up a wiring closet
- Troubleshooting a connection between two devices when a link cannot be established or when it is not operating properly

Running TDR and Displaying the Results

To run TDR, enter the **test cable-diagnostics tdr interface** *interface-id* privileged EXEC command:

To display the results, enter the **show cable-diagnostics tdr interface** *interface-id* privileged EXEC command. For a description of the fields in the display, see the command reference for this release.

**Note**

TDR is supported only on the copper Ethernet 10/100 ports or on dual-purpose ports configured as 10/100/100 ports by using the RJ-45 connector.

Using Debug Commands

- [Enabling Debugging on a Specific Feature, page 18-14](#)
- [Enabling All-System Diagnostics, page 18-14](#)
- [Redirecting Debug and Error Message Output, page 18-15](#)

**Caution**

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. It is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

**Note**

For complete syntax and usage information for specific **debug** commands, see the command reference for this release.

Enabling Debugging on a Specific Feature

All **debug** commands are entered in privileged EXEC mode, and most **debug** commands take no arguments. For example, beginning in privileged EXEC mode, enter this command to enable the debugging for Switched Port Analyzer (SPAN):

```
Switch# debug span-session
```

The switch module continues to generate output until you enter the **no** form of the command.

If you enable a **debug** command and no output appears, consider these possibilities:

- The switch module might not be properly configured to generate the type of traffic that you want to monitor. Use the **show running-config** command to check its configuration.
- Even if the switch module is properly configured, it might not generate the type of traffic that you want to monitor during the particular period that debugging is enabled. Depending on the feature you are debugging, you can use commands such as the TCP/IP **ping** command to generate network traffic.

To disable debugging of SPAN, enter this command in privileged EXEC mode:

```
Switch# no debug span-session
```

Alternately, in privileged EXEC mode, you can enter the **undebug** form of the command:

```
Switch# undebug span-session
```

To display the state of each debugging option, enter this command in privileged EXEC mode:

```
Switch# show debugging
```

Enabling All-System Diagnostics

Beginning in privileged EXEC mode, enter this command to enable all-system diagnostics:

```
Switch# debug all
```

**Caution**

Because debugging output takes priority over other network traffic, and because the **debug all** privileged EXEC command generates more output than any other **debug** command, it can severely diminish switch module performance or even render it unusable. In virtually all cases, it is best to use more specific **debug** commands.

The **no debug all** privileged EXEC command disables all diagnostic output. Using the **no debug all** command is a convenient way to ensure that you have not accidentally left any **debug** commands enabled.

Redirecting Debug and Error Message Output

By default, the network server sends the output from **debug** commands and system error messages to the console. If you use this default, you can use a virtual terminal connection to monitor debug output instead of connecting to the console port.

Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. The syslog format is compatible with 4.3 Berkeley Standard Distribution (BSD) UNIX and its derivatives.

**Note**

Be aware that the debugging destination you use affects system overhead. Logging messages to the console produces very high overhead, whereas logging messages to a virtual terminal produces less overhead. Logging messages to a syslog server produces even less, and logging to an internal buffer produces the least overhead of any method.

Using the *show platform forward* Command

The output from the **show platform forward** privileged EXEC command provides some useful information about the forwarding results if a packet entering an interface is sent through the system. Depending upon the parameters entered about the packet, the output provides lookup table results and port maps used to calculate forwarding destinations, bitmaps, and egress information.

**Note**

For more syntax and usage information for the **show platform forward** command, see the switch module command reference for this release.

Most of the information in the output from the command is useful mainly for technical support personnel, who have access to detailed information about the switch module ASICs. However, packet forwarding information can also be helpful in troubleshooting.

This is an example of the output from the **show platform forward** command on Gigabit Ethernet port 1 in VLAN 5 when the packet entering that port is addressed to unknown MAC addresses. The packet should be flooded to all other ports in VLAN 5.

```

Switch# show platform forward gigabitethernet0/1 vlan 5 1.1.1 2.2.2 ip 13.1.1.1 13.2.2.2
udp 10 20
Global Port Number:24, Asic Number:5
Src Real Vlan Id:5, Mapped Vlan Id:5

Ingress:
  Lookup                Key-Used                Index-Hit  A-Data
InptACL  40_0D020202_0D010101-00_40000014_000A0000    01FFA    03000000
L2Local  80_00050002_00020002-00_00000000_00000000    00C71    0000002B
Station Descriptor:02340000, DestIndex:0239, RewriteIndex:F005

=====
Egress:Asic 2, switch 1
Output Packets:

-----
Packet 1
  Lookup                Key-Used                Index-Hit  A-Data
OutptACL 50_0D020202_0D010101-00_40000014_000A0000    01FFE    03000000

Port      Vlan      SrcMac          DstMac          Cos  Dscp
Gi0/1     0005     0001.0001.0001  0002.0002.0002

-----
Packet 2
  Lookup                Key-Used                Index-Hit  A-Data
OutptACL 50_0D020202_0D010101-00_40000014_000A0000    01FFE    03000000

Port      Vlan      SrcMac          DstMac          Cos  Dscp
Gi0/2     0005     0001.0001.0001  0002.0002.0002

-----
<output truncated>
-----
Packet 10
  Lookup                Key-Used                Index-Hit  A-Data
OutptACL 50_0D020202_0D010101-00_40000014_000A0000    01FFE    03000000
Packet dropped due to failed DEJA_VU Check on Gi0/2

```

This is an example of the output when the packet coming in on Gigabit Ethernet port 1 in VLAN 5 is sent to an address already learned on the VLAN on another port. It should be forwarded from the port on which the address was learned.

```

Switch# show platform forward gigabitethernet0/1 vlan 5 1.1.1 0009.43a8.0145 ip 13.1.1.1
13.2.2.2 udp 10 20
Global Port Number:24, Asic Number:5
Src Real Vlan Id:5, Mapped Vlan Id:5

Ingress:
  Lookup                Key-Used                Index-Hit  A-Data
InptACL  40_0D020202_0D010101-00_40000014_000A0000    01FFA    03000000
L2Local  80_00050009_43A80145-00_00000000_00000000    00086    02010197
Station Descriptor:F0050003, DestIndex:F005, RewriteIndex:0003

=====
Egress:Asic 3, switch 1
Output Packets:

-----
Packet 1
  Lookup                Key-Used                Index-Hit  A-Data
OutptACL 50_0D020202_0D010101-00_40000014_000A0000    01FFE    03000000

Port      Vlan      SrcMac          DstMac          Cos  Dscp
Gi0/2     0005     0001.0001.0001  0009.43A8.0145

```


This is an example of the output when the packet coming in on Gigabit Ethernet port 1 in VLAN 5 has a destination MAC address set to the router MAC address in VLAN 5 and the destination IP address unknown. Because there is no default route set, the packet should be dropped.

```
Switch# show platform forward gigabitethernet0/1 vlan 5 1.1.1 03.e319.ee44 ip 13.1.1.1
13.2.2.2 udp 10 20
Global Port Number:24, Asic Number:5
Src Real Vlan Id:5, Mapped Vlan Id:5

Ingress:
  Lookup                Key-Used                Index-Hit  A-Data
InptACL  40_0D020202_0D010101-00_41000014_000A0000    01FFA  03000000
L3Local  00_00000000_00000000-90_00001400_0D020202        010F0  01880290
L3Scndr  12_0D020202_0D010101-00_40000014_000A0000        034E0  000C001D_00000000
Lookup Used:Secondary
Station Descriptor:02260000, DestIndex:0226, RewriteIndex:0000
```

This is an example of the output when the packet coming in on Gigabit Ethernet port 1 in VLAN 5 has a destination MAC address set to the router MAC address in VLAN 5 and the destination IP address set to an IP address that is in the IP routing table. It should be forwarded as specified in the routing table.

```
Switch# show platform forward gigabitethernet0/1 vlan 5 1.1.1 03.e319.ee44 ip 110.1.5.5
16.1.10.5
Global Port Number:24, Asic Number:5
Src Real Vlan Id:5, Mapped Vlan Id:5

Ingress:
  Lookup                Key-Used                Index-Hit  A-Data
InptACL  40_10010A05_0A010505-00_41000014_000A0000    01FFA  03000000
L3Local  00_00000000_00000000-90_00001400_10010A05        010F0  01880290
L3Scndr  12_10010A05_0A010505-00_40000014_000A0000        01D28  30090001_00000000
Lookup Used:Secondary
Station Descriptor:F0070007, DestIndex:F007, RewriteIndex:0007

=====
Egress:Asic 3, switch 1
Output Packets:

-----
Packet 1
  Lookup                Key-Used                Index-Hit  A-Data
OutptACL 50_10010A05_0A010505-00_40000014_000A0000    01FFE  03000000

Port      Vlan      SrcMac      DstMac      Cos  Dscpv
Gi0/2    0007  XXXX.XXXX.0246  0009.43A8.0147
```

Using the *crashinfo* File

The crashinfo file saves information that helps Cisco technical support representatives to debug problems that caused the Cisco IOS image to fail (crash). The switch module writes the crash information to the console at the time of the failure, and the file is created the next time you boot the Cisco IOS image after the failure (instead of while the system is failing).

The information in the file includes the Cisco IOS image name and version that failed, a list of the processor registers, and a stack trace. You can provide this information to the Cisco technical support representative by using the **show tech-support** privileged EXEC command.

All crashinfo files are kept in this directory on the flash file system:

flash:/crashinfo/crashinfo_*n* where *n* is a sequence number.

Each new crashinfo file that is created uses a sequence number that is larger than any previously existing sequence number, so the file with the largest sequence number describes the most recent failure. Version numbers are used instead of a timestamp because the switches do not include a real-time clock. You cannot change the name of the file that the system will use when it creates the file. However, after the file is created, you can use the **rename** privileged EXEC command to rename it, but the contents of the renamed file will not be displayed by the **show tech-support** privileged EXEC command. You can delete crashinfo files by using the **delete** privileged EXEC command.

You can display the most recent crashinfo file (that is, the file with the highest sequence number at the end of its filename) by entering the **show tech-support** privileged EXEC command. You also can access the file by using any command that can copy or display files, such as the **more** or the **copy** privileged EXEC command.

Using On-Board Failure Logging

You can use the on-board-failure logging (OBFL) feature to collect information about the switch module. The information includes uptime, temperature, and voltage information and helps Cisco technical support representatives to troubleshoot switch module problems.

This section has this information:

- [Understanding OBFL, page 18-18](#)
- [Configuring OBFL, page 18-19](#)
- [Displaying OBFL Information, page 18-19](#)

Understanding OBFL

By default, OBFL is enabled. It collects information about the switch module and small form-factor pluggable (SFP) modules. The switch module stores this information in the flash memory:

- CLI commands—Record of the OBFL CLI commands that are entered on a switch module
- Environmental data—Unique Device Identifier (UDI) information for a switch module and for all the connected devices: the product identification (PID), the version identification (VID), and the serial number
- Message—Record of the hardware-related system messages generated by a switch module
- Temperature—Temperature of a switch module
- Uptime data—Time when a switch module starts, the reason the switch module restarts, and the length of time the switch module has been running since it last restarted
- Voltage—System voltages of a switch module

You should manually set the system clock or configure it by using Network Time Protocol (NTP).

When the switch module is running, you can retrieve the OBFL data by using the **show logging onboard** privileged EXEC commands. If the switch module fails, contact your Cisco technical support representative to find out how to retrieve the data.

When an OBFL-enabled switch module is restarted, there is a 10-minute delay before logging of new data begins.

Configuring OBFL

To enable OBFL, use the **hw-module module logging onboard [message level level]** global configuration command. Use the **message level level** parameter to specify the severity of the hardware-related messages that the switch module generates and stores in the flash memory.

To copy the OBFL data to the local network or a specific file system, use the **copy logging onboard module 1 destination** privileged EXEC command.



Note We recommend that you keep OBFL enabled and that you do not remove the data stored in the flash memory.

Beginning in privileged EXEC mode, follow these steps to enable and configure OBFL. Note that OBFL is enabled by default; you need to enable it only if it has been disabled.

| | Step | Command |
|--------|--|---|
| Step 1 | Enter global configuration mode. | configure terminal |
| Step 2 | Enable OBFL on the switch module. You can specify these optional parameters: <ul style="list-style-type: none"> (Optional) <i>slot-number</i>—The slot number is always 1 and is not relevant for the switch module. (Optional) message level—Specify the severity level of messages to be generated and stored. The range is from 1 to 7, with 1 being the most severe. | hw-module module [slot-number] logging onboard [message level] |
| Step 3 | Return to privileged EXEC mode. | end |
| Step 4 | (Optional) Copy the OBFL data to the local network or a specific file system. <ul style="list-style-type: none"> (Optional) <i>slot-number</i>—The slot number is always 1 and is not relevant for the switch module. <i>destination</i>—See the copy logging onboard module command for destination options. | copy logging onboard module [slot-number] destination |
| Step 5 | Verify your entries. | show logging onboard |
| Step 6 | (Optional) Save your entries in the configuration file. | copy running-config startup-config |

To disable OBFL, use the no **hw-module module 1 logging onboard [message level]** global configuration command.

To clear all the OBFL data in the flash memory except for the uptime and CLI command information, use the **clear logging onboard** privileged EXEC command.

For more information about the commands in this section, see the command reference for this release.

Displaying OBFL Information

To display the OBFL information, use one or more of the privileged EXEC commands in [Table 18-3](#).

**Note**

When an OBFL-enabled switch module is restarted, there is a 10-minute delay before logging of new data begins.

Table 18-3 *Commands for Displaying OBFL Information*

| Command | Description |
|---|--|
| show logging onboard clilog | Displays the OBFL CLI commands that were entered on a switch module |
| show logging onboard environment | Displays the UDI information for a standalone switch module and for all the connected FRU devices: the PID, the VID, and the serial number |
| show logging onboard message | Displays the hardware-related messages generated by a switch module |
| show logging onboard temperature | Displays the temperature of a switch module |

These are examples of output from the show logging onboard commands:

```

Switch# show logging onboard cliilog
-----
CLI LOGGING SUMMARY INFORMATION
-----
COUNT COMMAND
-----
  1 hw-module module logging onboard
  1 hw-module module logging onboard message level 7
  4 show logging onboard
  1 show logging onboard message
  1 show logging onboard summary
-----

Switch# show logging onboard temp
-----
TEMPERATURE SUMMARY INFORMATION
-----
Number of sensors          : 1
Sampling frequency        : 5 minutes
Maximum time of storage    : 720 minutes
-----
Sensor                    | ID | Maximum Temperature 0C
-----
System                    | 1  | 41
-----
Temp                      |    | Sensor ID
0C                        | 1  |
-----
No historical data to display
-----

Switch# show logging onboard uptime
-----
UPTIME SUMMARY INFORMATION
-----
First customer power on : 03/01/1993 00:06:06
Total uptime           : 0 years 20 weeks 4 days 6 hours 20 minutes
Total downtime         : 0 years 0 weeks 0 days 0 hours 0 minutes
Number of resets       : 90
Number of slot changes : 0
Current reset reason    : 0x0
Current reset timestamp : 03/01/1993 00:05:43
Current slot           : 1
Current uptime         : 0 years 0 weeks 2 days 6 hours 0 minutes
-----
Reset |          |
Reason | Count  |
-----
No historical data to display
-----

```

```
Switch# show logging onboard voltage
```

```
-----  
VOLTAGE SUMMARY INFORMATION  
-----
```

```
Number of sensors      : 6  
Sampling frequency    : 1 minutes  
Maximum time of storage : 720 minutes  
-----
```

| Sensor | ID | Maximum Voltage |
|--------|----|-----------------|
| 12.00V | 0 | 12.567 |
| 1.25V | 2 | 1.258 |
| 3.30V | 3 | 3.305 |
| 2.50V | 4 | 2.517 |
| 1.80V | 5 | 1.825 |
| 1.50V | 6 | 1.508 |

```
-----
```

```
Nominal Range          Sensor ID
```

```
-----  
No historical data to display  
-----
```