



# Tracing and Trace Management

---

- [Tracing Overview, on page 1](#)
- [How Tracing Works, on page 1](#)
- [Tracing Levels, on page 2](#)
- [Viewing a Tracing Level, on page 3](#)
- [Setting a Tracing Level, on page 5](#)
- [Viewing the Content of the Trace Buffer, on page 5](#)

## Tracing Overview

Tracing is a function that logs internal events. Trace files are automatically created and saved to the `tracelogs` directory on the harddisk: file system on the router, which stores tracing files in `bootflash:`. Trace files are used to store tracing data.

The contents of trace files are useful for the following purposes:

- **Troubleshooting**—If a router is having an issue, the trace file output may provide information that is useful for locating and solving the problem. Trace files can almost always be accessed through diagnostic mode even if other system issues are occurring.
- **Debugging**—The trace file outputs can help users get a more detailed view of system actions and operations.

## How Tracing Works

The tracing function logs the contents of internal events on the router. Trace files with all trace output for a module are periodically created and updated and are stored in the `tracelog` directory. Trace files can be erased from this directory to recover space on the file system without impacting system performance.

The most recent trace information for a specific module can be viewed using the **`show platform software trace message`** privileged EXEC and diagnostic mode command. This command can be entered to gather trace log information even during an IOS failure because it is available in diagnostic mode.

Trace files can be copied to other destinations using most file transfer functions (such as FTP, TFTP, and so on) and opened using a plaintext editor.

Tracing cannot be disabled on the router. Trace levels, however, which set the message types that generate trace output, are user-configurable and can be set using the **`set platform software trace`** command. If a user

wants to modify the trace level to increase or decrease the amount of trace message output, the user should set a new tracing level using the **set platform software trace** command. Trace levels can be set by process using the **all-modules** keyword within the **set platform software trace** command, or by module within a process. See the **set platform software trace** command reference for more information on this command, and the [Tracing Levels, on page 2](#) section of this document for additional information on tracing levels.

## Tracing Levels

Tracing levels determine how much information about a module should be stored in the trace buffer or file.

The table below shows all of the trace levels that are available and provides descriptions of what types of messages are displayed with each tracing level.

**Table 1: Tracing Levels and Descriptions**

Trace Level	Level Number	Description
Emergency	0	The message is regarding an issue that makes the system unusable.
Alert	1	The message is regarding an action that must be taken immediately.
Critical	2	The message is regarding a critical condition. This is the default setting.
Error	3	The message is regarding a system error.
Warning	4	The message is regarding a system warning
Notice	5	The message is regarding a significant issue, but the router is still working normally.
Informational	6	The message is useful for informational purposes only.
Debug	7	The message provides debug-level output.
Verbose	8	All possible tracing messages are sent.
Noise	-	All possible trace messages for the module are logged.  The noise level is always equal to the highest possible tracing level. Even if a future enhancement to tracing introduces a higher tracing level, the noise level will become equal to the level of that new enhancement.

Trace level settings are leveled, meaning that every setting will contain all messages from the lower setting plus the messages from its own setting. For instance, setting the trace level to 3(error) ensures that the trace file will contain all output for the 0 (emergencies), 1 (alerts), 2 (critical), and 3 (error) settings. Setting the trace level to 4 (warning) will ensure that all trace output for the specific module will be included in that trace file.

The default tracing level for every module on the router is notice.

All trace levels are not user-configurable. Specifically, the alert, critical, and notice tracing levels cannot be set by users. If you wish to trace these messages, set the trace level to a higher level that will collect these messages.

When setting trace levels, it is also important to remember that the setting is not done in a configuration mode, so trace level settings are returned to their defaults after every router reload.



**Caution** Setting tracing of a module to the debug level or higher can have a negative performance impact. Setting tracing to this level or higher should be done with discretion.



**Caution** Setting a large number of modules to high tracing levels can severely degrade performance. If a high level of tracing is needed in a specific context, it is almost always preferable to set a single module on a higher tracing level rather than setting multiple modules to high tracing levels.

## Viewing a Tracing Level

By default, all modules on the router are set to notice. This setting will be maintained unless changed by a user.

To see the tracing level for any module on the router, enter the **show platform software trace level** command in privileged EXEC or diagnostic mode.

In the following example, the **show platform software trace level** command is used to view the tracing levels of the Forwarding Manager processes:

```
Router#show platform software trace level forwarding-manager rp active
Module Name                               Trace Level
-----
acl                                         Notice
bfd                                         Notice
binos                                       Notice
bipc                                        Notice
bridge-domain                             Notice
bsignal                                    Notice
btrace                                     Notice
bump_ptr_alloc                             Notice
cce                                         Notice
cdllib                                     Notice
cef                                         Notice
chasfs                                      Notice
chasutil                                   Notice
cos-marking                                Notice
cyan                                        Notice
efp                                         Notice
eoam                                        Notice
ether-channel                              Notice
ether-dplb                                 Notice
evlib                                       Notice
evutil                                     Notice
fhrp                                        Notice
file_alloc                                 Notice
flash                                       Notice
fman_rp                                    Notice
inject-marking                             Notice
interfaces                                  Notice
ipc                                         Notice
ipclog                                     Notice
ipp-to-cos                                 Notice
```

ipsec	Notice
ipsla	Notice
l2cp	Notice
l2fib	Notice
local_span	Notice
macinmac	Notice
mgmte-acl	Notice
mqipc	Notice
om	Notice
pbr	Notice
peer	Notice
protection	Notice
protocol-marking	Notice
punt-police	Notice
qos	Notice
qos-account	Notice
qos-event	Notice
qos-hqf	Notice
qos-infra	Notice
qos-init	Notice
qos-police	Notice
qos-set	Notice
qos-stats	Notice
remote_span	Notice
route-map	Notice
services	Notice
source	Notice
subsys	Notice
sw_wdog	Notice
syshw	Notice
tcl_acl_db	Notice
tcl_bdomain_common	Notice
tcl_bdomain_db	Notice
tcl_cdlcore	Notice
tcl_cef_config	Notice
tcl_cef_config_common	Notice
tcl_cef_route	Notice
tcl_dpibd_config	Notice
tcl_dpibd_db	Notice
tcl_ether_efp	Notice
tcl_ether_efp_db	Notice
tcl_fman_rp_uea	Notice
tcl_ipc_ack	Notice
tcl_l2cp_db	Notice
tcl_l2fib_config	Notice
tcl_l2fib_db	Notice
tcl_om	Notice
tcl_tcl_toc	Notice
tcl_ui	Notice
tcl_urpf_config	Notice
tcl_urpf_db	Notice
tcl_vrf_config	Notice
tcl_vrf_db	Notice
tdllib	Notice
trans_avl	Notice
trans_gbt	Notice
trccfg	Notice
uihandler	Notice
uipeer	Notice
uistatus	Notice
urpf	Notice
virtual-ethernet	Notice
vista	Notice
vs_flock	Notice

## Setting a Tracing Level

To set a tracing level for any module on the router, or for all modules within a process on the router, enter the **set platform software trace** privileged EXEC and diagnostic mode command.

In the following example, the trace level for the ACL module in the Forwarding Manager of the ESP processor in slot 0 is set to info.

```
set platform software trace forwarding-manager F0 acl info
```

See the **set platform software trace** command reference for additional information about the options for this command.

## Viewing the Content of the Trace Buffer

To view the trace messages in the trace buffer or file, enter the **show platform software trace message** privileged EXEC and diagnostic mode command.

In the following example, the trace messages for the Host Manager process in Route Switch Processor slot 0 are viewed using the **show platform software trace message** command:

```
Router# show platform software trace message host-manager R0
08/23 12:09:14.408 [uipeer]: (info): Looking for a ui_req msg
08/23 12:09:14.408 [uipeer]: (info): Start of request handling for con 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Accepted connection for 14 as 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Received new connection 0x100a61c8 on descriptor 14
08/23 12:09:14.398 [uipeer]: (info): Accepting command connection on listen fd 7
08/23 11:53:57.440 [uipeer]: (info): Going to send a status update to the shell manager in
slot 0
08/23 11:53:47.417 [uipeer]: (info): Going to send a status update to the shell manager in
slot 0
```

