



## Debug Commands: j to q

---

- [debug l2age](#), on page 2
- [debug mac](#), on page 3
- [debug mdns all](#), on page 4
- [debug mdns detail](#), on page 5
- [debug mdns error](#), on page 6
- [debug mdns message](#), on page 7
- [debug mdns ha](#), on page 8
- [debug memory](#), on page 9
- [debug mesh security](#), on page 10
- [debug mesh convergence](#), on page 11
- [debug mobility](#), on page 12
- [debug nac](#), on page 14
- [debug nmsp](#), on page 15
- [debug ntp](#), on page 16
- [debug packet error](#), on page 17
- [debug packet logging](#), on page 18
- [debug pem](#), on page 21
- [debug pm](#), on page 22
- [debug poe](#), on page 24
- [debug policy](#), on page 25
- [debug profiling](#), on page 26

# debug l2age

To configure the debugging of Layer 2 age timeout messages, use the **debug l2age** command.

```
debug l2age { enable | disable }
```

## Syntax Description

<b>enable</b>	Enables the debugging of Layer2 age settings.
<b>disable</b>	Disables the debugging Layer2 age settings.

## Command Default

None

## Command History

### Release Modification

7.6	This command was introduced in a release earlier than Release 7.6.
-----	--

The following example shows how to enable the debugging of Layer2 age settings:

```
(Cisco Controller) > debug l2age enable
```

## Related Commands

**debug disable-all**

# debug mac

To configure the debugging of the client MAC address, use the **debug mac** command.

```
debug mac {disable | addr MAC}
```

Syntax Description	Parameter	Description
	<b>disable</b>	Disables the debugging of the client using the MAC address.
	<b>addr</b>	Configures the debugging of the client using the MAC address.
	<i>MAC</i>	MAC address of the client.

**Command Default** None

Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to configure the debugging of the client using the MAC address:

```
(Cisco Controller) > debug mac addr 00.0c.41.07.33.a6
```

**Related Commands** [debug disable-all](#)

# debug mdns all

To debug all multicast DNS (mDNS) messages, details, and errors, use the **debug mdns all** command.

```
debug mdns all { enable | disable }
```

## Syntax Description

**enable** Enables the debugging of all mDNS messages, details, and errors.

**disable** Disables the debugging of all mDNS messages, details, and errors.

## Command Default

By default, the debugging of all mDNS messages, details, and errors is disabled.

## Command History

### Release Modification

7.4 This command was introduced.

The following example shows how to enable debugging of all mDNS messages, details, and errors:

```
(Cisco Controller) > debug mdns all enable
```

## Related Commands

- config mdns profile**
- config mdns query interval**
- config mdns service**
- config mdns snooping**
- config interface mdns-profile**
- config interface group mdns-profile**
- config wlan mdns**
- show mdns profile**
- show mnds service**
- clear mdns service-database**
- debug mdns error**
- debug mdns detail**

## debug mdns detail

To debug multicast DNS (mDNS) details, use the **debug mdns detail** command.

**debug mdns detail** {enable | disable}

Syntax Description	
<b>enable</b>	Enables the debugging of mDNS details.
<b>disable</b>	Disables the debugging of mDNS details.

**Command Default** This command is disabled by default.

Command History	Release	Modification
	7.4	This command was introduced.

The following example shows how to enable the debugging of mDNS details:

```
(Cisco Controller) > debug mdns detail enable
```

Related Commands	
	<b>config mdns profile</b>
	<b>config mdns query interval</b>
	<b>config mdns service</b>
	<b>config mdns snooping</b>
	<b>config interface mdns-profile</b>
	<b>config interface group mdns-profile</b>
	<b>config wlan mdns</b>
	<b>show mdns profile</b>
	<b>show mnds service</b>
	<b>clear mdns service-database</b>
	<b>debug mdns all</b>
	<b>debug mdns error</b>

## debug mdns error

To debug multicast DNS (mDNS) errors, use the **debug mdns error** command.

**debug mdns error** { **enable** | **disable** }

Syntax Description	
<b>enable</b>	Enables the debugging of mDNS errors.
<b>disable</b>	Disables the debugging of mDNS errors.

**Command Default** This command is disabled by default.

Command History	Release	Modification
	7.4	This command was introduced.

The following example shows how to enable the debugging of mDNS errors.

```
(Cisco Controller) > debug mdns error enable
```

Related Commands	
	<b>config mdns profile</b>
	<b>config mdns query interval</b>
	<b>config mdns service</b>
	<b>config mdns snooping</b>
	<b>config interface mdns-profile</b>
	<b>config interface group mdns-profile</b>
	<b>config wlan mdns</b>
	<b>show mdns profile</b>
	<b>show mnds service</b>
	<b>clear mdns service-database</b>
	<b>debug mdns all</b>
	<b>debug mdns detail</b>
	<b>debug mdns message</b>

# debug mdns message

To debug multicast DNS (mDNS) messages, use the **debug mdns message** command.

**debug mdns message** {enable | disable}

<b>Syntax Description</b>	<b>enable</b> Enables the debugging of mDNS messages.
	<b>disable</b> Disables the debugging of mDNS messages.

**Command Default** Disabled.

<b>Command History</b>	<b>Release</b> <b>Modification</b>
	7.4 This command was introduced.

The following example shows how to enable the debugging of mDNS messages:

```
(Cisco Controller) > debug mdns message enable
```

<b>Related Commands</b>	<b>config mdns profile</b>
	<b>config mdns query interval</b>
	<b>config mdns service</b>
	<b>config mdns snooping</b>
	<b>config interface mdns-profile</b>
	<b>config interface group mdns-profile</b>
	<b>config wlan mdns</b>
	<b>show mdns profile</b>
	<b>show mnds service</b>
	<b>clear mdns service-database</b>
	<b>debug mdns all</b>
	<b>debug mdns error</b>
	<b>debug mdns detail</b>

## debug mdns ha

To debug all the multicast Domain Name System (mDNS) High Availability (HA) messages, use the **debug mdns ha** command.

**debug mdns ha** { **enable** | **disable** }

<b>Syntax Description</b>	<b>enable</b> Enables debugging of all the mDNS HA messages.
	<b>disable</b> Disables debugging of all the mDNS HA messages.

**Command Default** This command is disabled by default.

<b>Command History</b>	<b>Release</b> <b>Modification</b>
	7.5 This command was introduced.

**Usage Guidelines** This command is automatically enabled when the **debug mdns all** command is enabled.

The following example shows how to enable debugging of all the mDNS HA messages:

```
(Cisco Controller) > debug mdns ha enable
```



# debug memory

To enable or disable the debugging of errors or events during the memory allocation of the controller, use the **debug memory** command.

**debug memory** { **errors** | **events** } { **enable** | **disable** }

Syntax Description	errors	events
	Configures the debugging of memory leak errors.	Configures debugging of memory leak events.
	enable	enable
	Disables the debugging of memory leak errors.	Disables the debugging of memory leak events.

**Command Default** By default, the debugging of errors or events during the memory allocation of the controller is disabled.

Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of memory leak events:

```
(Cisco Controller) > debug memory events enable
```

Related Commands
<ul style="list-style-type: none"> <li>config memory monitor errors</li> <li>show memory monitor</li> <li>config memory monitor leaks</li> </ul>

## debug mesh security

To configure the debugging of mesh security issues, use the **debug mesh security** command.

**debug mesh security** {**all** | **events** | **errors**} {**enable** | **disable**}

### Syntax Description

<b>all</b>	Configures the debugging of all mesh security messages.
<b>events</b>	Configures the debugging of mesh security event messages.
<b>errors</b>	Configures the debugging of mesh security error messages.
<b>enable</b>	Enables the debugging of mesh security error messages.
<b>disable</b>	Disables the debugging of mesh security error messages.

### Command Default

None

### Command History

Release	Modification
7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of mesh security error messages:

```
(Cisco Controller) >debug mesh security errors enable
```

# debug mesh convergence

To configure the debugging of mesh convergence issues, use the **debug mesh convergence** command.

## **debug mesh convergence**

---

**Syntax Description** This command has no arguments or keywords.

---

**Command Default** None

---

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	8.0	This command was introduced.

---

The following example shows how to enable the debugging of mesh coverage error messages:

```
(Cisco Controller) >debug mesh convergence  
mesh convergence debugging is on
```

## debug mobility

To configure the debugging of wireless mobility, use the **debug mobility** command.

**debug mobility** {**ap-list** | **config** | **directory** | **dtls** | **handoff** | **keep-alive** | **multicast** | **oracle** | **packet** | **peer-ip** *IP-address* | **pmk** | **pmtu-discovery** | **redha**} {**enable** | **disable**}

Syntax	Description
<b>ap-list</b>	Configures the debugging of wireless mobility access point list.
<b>config</b>	Configures the debugging of wireless mobility configuration.
<b>directory</b>	Configures the debugging of wireless mobility error messages.
<b>dtls</b>	Configures the debugging of wireless mobility Datagram Transport Layer Security (DTLS) options.
<b>handoff</b>	Configures the debugging of wireless mobility handoff messages.
<b>keep-alive</b>	Configures the debugging of wireless mobility CAPWAP data DTLS keep-alive packets.
<b>multicast</b>	Configures the debugging of multicast mobility packets.
<b>oracle</b>	Starts the debugging of wireless mobility oracle options.
<b>packet</b>	Configures the debugging of wireless mobility packets.
<b>peer-ip</b>	Configures IP address of the mobility peer for which incoming and outgoing mobility messages should be displayed.
<i>IP-address</i>	IP address of the mobility peer for which incoming and outgoing mobility messages should be displayed.
<b>pmk</b>	Configures the debugging of wireless mobility pairwise master key (PMK).
<b>pmtu-discovery</b>	Configures the debugging of the wireless mobility path MTU discovery.
<b>redha</b>	Configures the debugging of the multicast mobility high availability.

<b>enable</b>	Enables the debugging of the wireless mobility feature.
<b>disable</b>	Disables the debugging of the wireless mobility feature.

**Command Default**

None

**Command History**

<b>Release</b>	<b>Modification</b>
7.6	This command was introduced in a release earlier than Release 7.6.
8.0	This command supports both IPv4 and IPv6 address formats.

The following example shows how to enable the debugging of wireless mobility packets.

```
(Cisco Controller) >debug mobility handoff enable
```

# debug nac

To configure the debugging of Network Access Control (NAC), use the **debug nac** command.

**debug nac** { **events** | **packet** } { **enable** | **disable** }

Syntax Description		
<b>events</b>	Configures the debugging of NAC events.	
<b>packet</b>	Configures the debugging of NAC packets.	
<b>enable</b>	Enables the NAC debugging.	
<b>disable</b>	Disables the NAC debugging.	
Command Default	None	
Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of NAC settings:

```
(Cisco Controller) > debug nac events enable
```

Related Commands	
	<b>show nac statistics</b>
	<b>show nac summary</b>
	<b>config guest-lan nac</b>
	<b>config wlan nac</b>

## debug nmsp

To configure the debugging of the Network Mobility Services Protocol (NMSP), use the **debug nmsp** command.

**debug nmsp** { **all** | **connection** | **detail** | **error** | **event** | **message** | **packet** }

Syntax Description		
	<b>all</b>	Configures the debugging for all NMSP messages.
	<b>connection</b>	Configures the debugging for NMSP connection events.
	<b>detail</b>	Configures the debugging for NMSP events in detail.
	<b>error</b>	Configures the debugging for NMSP error messages.
	<b>event</b>	Configures the debugging for NMSP events.
	<b>message</b>	Configures the debugging for NMSP transmit and receive messages.
	<b>packet</b>	Configures the debugging for NMSP packet events.

**Command Default** None

Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to configure the debugging of NMSP connection events:

```
(Cisco Controller) > debug nmsp connection
```

**Related Commands**

- clear nmsp statistics
- debug disable-all
- config nmsp notify-interval measurement

# debug ntp

To configure the debugging of the Network Time Protocol (NTP), use the **debug ntp** command.

**debug ntp** { **detail** | **low** | **packet** } { **enable** | **disable** }

Syntax Description		
	<b>detail</b>	Configures the debugging of detailed NTP messages.
	<b>low</b>	Configures the debugging of NTP messages.
	<b>packet</b>	Configures the debugging of NTP packets.
	<b>enable</b>	Enables the NTP debugging.
	<b>disable</b>	Disables the NTP debugging.

**Command Default** None

Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of NTP settings:

```
(Cisco Controller) > debug ntp packet enable
```

**Related Commands** **debug disable-all**



## debug packet error

To configure debugging of the packets sent to the controller CPU , use the **debug packet error** command.

**debug packet error** {**enable** | **disable**}

---

**Syntax Description**

**enable** Enables debugging of the packets sent to the controller CPU.

**disable** Disables debugging of the packets sent to the controller CPU.

---

---

**Command Default**

None

---

**Command History**

---

**Release Modification**

7.6 This command was introduced in a release earlier than Release 7.6.

---

The following example shows how to enable the debugging of the packets sent to the controller CPU:

```
(Cisco Controller) > debug packet error enable
```

## debug packet logging

To configure logging of the packets sent to the controller CPU, use the **debug packet logging** command.

```
debug packet logging {acl | disable | enable {rx | tx | all} packet_count display_size | format {hex2pcap | text2pcap}}
```

```
debug packet logging acl {clear-all | driver rule_index action npu_encap port | eoip-eth rule_index action dst src type vlan | eoip-ip rule_index action src dst proto src_port dst_port | eth rule_index action dst src type vlan | ip rule_index action src dst proto src_port dst_port | lwapp-dot11rule_index action dst src bssid type | lwapp-ip rule_index action src dst proto src_port dst_port}
```

Syntax Description		
<b>acl</b>		Filters the displayed packets according to a rule.
<b>disable</b>		Disables logging of all the packets.
<b>enable</b>		Enables logging of all the packets.
<b>rx</b>		Displays all the received packets.
<b>tx</b>		Displays all the transmitted packets.
<b>all</b>		Displays both the transmitted and the received packets.
<i>packet_count</i>		Maximum number of packets to be logged. The range is from 1 to 65535. The default value is 25.
<i>display_size</i>		Number of bytes to be displayed when printing a packet. By default, the entire packet is displayed.
<b>format</b>		Configures the format of the debug output.
<b>hex2pcap</b>		Configures the output format to be compatible with the hex2pcap format. The standard format used by Cisco IOS supports the use of hex2pcap and can be decoded using an HTML front end.
<b>text2pcap</b>		Configures the output format to be compatible with the text2pcap format. In this format, the sequence of packets can be decoded from the same console log file. .
<b>clear-all</b>		Clears all the existing rules pertaining to the packets.
<b>driver</b>		Filters the packets based on an incoming port or a Network Processing Unit (NPU) encapsulation type.
<i>rule_index</i>		Index of the rule that is a value between 1 and 6 (inclusive).
<i>action</i>		Action for the rule, which can be <b>permit</b> , <b>deny</b> , or <b>disable</b> .

<i>npu_encap</i>	NPU encapsulation type that determines how the packets are filtered. The possible values are <i>dhcp</i> , <i>dot11-mgmt</i> , <i>dot11-probe</i> , <i>dot1x</i> , <i>eoip-ping</i> , <i>iapp</i> , <i>ip</i> , <i>lwapp</i> , <i>multicast</i> , <i>orphan-from-sta</i> , <i>orphan-to-sta</i> , <i>rbcf</i> , <i>wired-guest</i> , or <i>any</i> .
<i>port</i>	Physical port for packet transmission or reception.
<b>eoip-eth</b>	Filters packets based on the Ethernet II header in the Ethernet over IP (EoIP) payload.
<i>dst</i>	Destination MAC address.
<i>src</i>	Source MAC address.
<i>type</i>	Two-byte type code, such as 0x800 for IP, 0x806 for Address Resolution Protocol (ARP). You can also enter a few common string values such as <i>ip</i> (for 0x800) or <i>arp</i> (for 0x806).
<i>vlan</i>	Two-byte VLAN identifier.
<b>eoip-ip</b>	Filters packets based on the IP header in the EoIP payload.
<i>proto</i>	Protocol. Valid values are: <i>ip</i> , <i>icmp</i> , <i>igmp</i> , <i>ggp</i> , <i>ipencap</i> , <i>st</i> , <i>tcp</i> , <i>egp</i> , <i>pup</i> , <i>udp</i> , <i>hmp</i> , <i>xns-idp</i> , <i>rdp</i> , <i>iso-tp4</i> , <i>xtp</i> , <i>ddp</i> , <i>idpr-cmtp</i> , <i>rspf</i> , <i>vmtp</i> , <i>ospf</i> , <i>ipip</i> , and <i>encap</i> .
<i>src_port</i>	User Datagram Protocol or Transmission Control Protocol (UDP or TCP) two-byte source port, such as <i>telnet</i> , <i>23</i> , or <i>any</i> . The controller supports the following strings: <i>tcpmux</i> , <i>echo</i> , <i>discard</i> , <i>systat</i> , <i>daytime</i> , <i>netstat</i> , <i>qotd</i> , <i>misp</i> , <i>chargen</i> , <i>ftp-data</i> , <i>ftp</i> , <i>fsp</i> , <i>ssh</i> , <i>telnet</i> , <i>smtp</i> , <i>time</i> , <i>rlp</i> , <i>nameserver</i> , <i>whois</i> , <i>re-mail-ck</i> , <i>domain</i> , <i>mtp</i> , <i>bootps</i> , <i>bootpc</i> , <i>tftp</i> , <i>gopher</i> , <i>rje</i> , <i>finger</i> , <i>www</i> , <i>link</i> , <i>kerberos</i> , <i>supdup</i> , <i>hostnames</i> , <i>iso-tsap</i> , <i>csnet-ns</i> , <i>3com-tsmux</i> , <i>rtnet</i> , <i>pop-2</i> , <i>pop-3</i> , <i>sunrpc</i> , <i>auth</i> , <i>sftp</i> , <i>uucp-path</i> , <i>nntp</i> , <i>ntp</i> , <i>netbios-ns</i> , <i>netbios-dgm</i> , <i>netbios-ssn</i> , <i>imap2</i> , <i>snmp</i> , <i>snmp-trap</i> , <i>cmip-man</i> , <i>cmip-agent</i> , <i>xmcp</i> , <i>nextstep</i> , <i>bgp</i> , <i>prospero</i> , <i>irc</i> , <i>smux</i> , <i>at-rtmp</i> , <i>at-nbp</i> , <i>at-echo</i> , <i>at-zis</i> , <i>qntp</i> , <i>z3950</i> , <i>ipx</i> , <i>imap3</i> , <i>ulistserv</i> , <i>https</i> , <i>snpp</i> , <i>saft</i> , <i>npmp-local</i> , <i>npmp-gui</i> , and <i>hmmmp-ind</i> .
<i>dst_port</i>	UDP or TCP two-byte destination port, such as <i>telnet</i> , <i>23</i> , or <i>any</i> . The controller supports the same strings as those for the <i>src_port</i> .
<b>eth</b>	Filters packets based on the values in the Ethernet II header.
<b>ip</b>	Filters packets based on the values in the IP header.
<b>lwapp-dot11</b>	Filters packets based on the 802.11 header in the Lightweight Access Point Protocol (LWAPP) payload.
<i>bssid</i>	Basic Service Set Identifier of the VLAN.
<b>lwapp-ip</b>	Filters packets based on the IP header in the LWAPP payload.

---

<b>Command Default</b>	None
------------------------	------

---

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	7.6	This command was introduced in a release earlier than Release 7.6.

---

The following example shows how to enable logging of a packet:

```
(Cisco Controller) > debug packet logging enable
```

# debug pem

To configure debugging of the access policy manager, use the **debug pem** command.

**debug pem** {events | state} {enable | disable}

Syntax Description	events	Configures the debugging of the policy manager events.
	state	Configures the debugging of the policy manager state machine.
	enable	Enables the debugging of the access policy manager.
	disable	Disables the debugging of the access policy manager.
Command Default	None	
Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of the access policy manager:

```
(Cisco Controller) >debug pem state enable
```

# debug pm

To configure the debugging of the security policy manager module, use the **debug pm** command.

```
debug pm {all disable | {config | hwcrypto | ikemsg | init | list | message | pki | rng
| rules | sa-export | sa-import | ssh-l2tp | ssh-appgw | ssh-engine | ssh-int | ssh-pmgr
| ssh-ppp | ssh-tcp} {enable | disable}}
```

## Syntax Description

<b>all disable</b>	Disables all debugging in the policy manager module.
<b>config</b>	Configures the debugging of the policy manager configuration.
<b>hwcrypto</b>	Configures the debugging of hardware offload events.
<b>ikemsg</b>	Configures the debugging of Internet Key Exchange (IKE) messages.
<b>init</b>	Configures the debugging of policy manager initialization events.
<b>list</b>	Configures the debugging of policy manager list mgmt.
<b>message</b>	Configures the debugging of policy manager message queue events.
<b>pki</b>	Configures the debugging of Public Key Infrastructure (PKI) related events.
<b>rng</b>	Configures the debugging of random number generation.
<b>rules</b>	Configures the debugging of Layer 3 policy events.
<b>sa-export</b>	Configures the debugging of SA export (mobility).
<b>sa-import</b>	Configures the debugging of SA import (mobility).
<b>ssh-l2tp</b>	Configures the debugging of policy manager Layer 2 Tunneling Protocol (L2TP) handling.
<b>ssh-appgw</b>	Configures the debugging of application gateways.
<b>ssh-engine</b>	Configures the debugging of the policy manager engine.
<b>ssh-int</b>	Configures the debugging of the policy manager interceptor.
<b>ssh-pmgr</b>	Configures the debugging of the policy manager.

<b>ssh-ppp</b>	Configures the debugging of policy manager Point To Point Protocol (PPP) handling.
<b>ssh-tcp</b>	Configures the debugging of policy manager TCP handling.
<b>enable</b>	Enables the debugging.
<b>disable</b>	Disables the debugging.

**Command Default** None

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to configure the debugging of PKI-related events:

```
(Cisco Controller) > debug pm pki enable
```

**Related Commands** **debug disable-all**

# debug poe

To configure the debugging of Power over Ethernet (PoE), use the **debug poe** command.

```
debug poe {detail | message | error} {enable | disable}
```

## Syntax Description

<b>detail</b>	Configures the debugging of PoE detail logs.
<b>error</b>	Configures the debugging of PoE error logs.
<b>message</b>	Configures the debugging of PoE messages.
<b>enable</b>	Enables the debugging of PoE logs.
<b>disable</b>	Disables the debugging of PoE logs.

## Command Default

None

## Command History

### Release Modification

7.6 This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the PoE debugging:

```
(Cisco Controller) > debug poe message enable
```

## Related Commands

**debug disable-all**



# debug policy

To configure debugging of policy settings, use the **debug policy** command.

**debug policy** {**errors** | **events**} {**enable** | **disable**}

Syntax Description		
	<b>errors</b>	Configures debugging of policy errors.
	<b>events</b>	Configures debugging of policy events.
	<b>enable</b>	Enables debugging of policy events.
	<b>disable</b>	Disables debugging of policy events.
Command Default	None	
Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable debugging of policy errors:

```
(Cisco Controller) > debug policy errors enable
```

## debug profiling

To configure the debugging of client profiling, use the **debug profiling** command.

**debug profiling** { **enable** | **disable** }

Syntax Description	
<b>enable</b>	Enables the debugging of client profiling (HTTP and DHCP profiling).
<b>disable</b>	Disables the debugging of client profiling (HTTP and DHCP profiling).

Command Default	
	Disabled.

Command History	Release	Modification
	7.6	This command was introduced in a release earlier than Release 7.6.

The following example shows how to enable the debugging of client profiling:

```
(Cisco Controller) >debug profiling enable
```