



## Debug Commands: a to i

---

- [debug aaa](#), on page 3
- [debug aaa events](#), on page 4
- [debug aaa local-auth](#), on page 5
- [debug airewave-director](#), on page 7
- [debug ap](#), on page 9
- [debug ap enable](#), on page 11
- [debug ap packet-dump](#), on page 12
- [debug ap show stats](#), on page 13
- [debug ap show stats video](#), on page 15
- [debug arp](#), on page 16
- [debug avc](#), on page 17
- [debug bcast](#), on page 18
- [debug call-control](#), on page 19
- [debug capwap](#), on page 20
- [debug capwap reap](#), on page 21
- [debug ccxdiag](#), on page 22
- [debug ccxrm](#), on page 23
- [debug ccxs69](#), on page 24
- [debug cckm](#), on page 25
- [debug client](#), on page 26
- [debug cts aaa](#), on page 27
- [debug cts authz](#), on page 28
- [debug cts capwap](#), on page 29
- [debug cts env-data](#), on page 30
- [debug cts ha](#), on page 31
- [debug cts key-store](#), on page 32
- [debug cts provisioning](#), on page 33
- [debug cts sgt](#), on page 34
- [debug cts sxp](#), on page 35
- [debug cac](#), on page 36
- [debug cdp](#), on page 37
- [debug crypto](#), on page 38
- [debug dhcp](#), on page 39

- [debug dhcp service-port](#), on page 40
- [debug disable-all](#), on page 41
- [debug dns](#), on page 42
- [debug dot11](#), on page 43
- [debug dot11](#), on page 44
- [debug dot11 mgmt interface](#), on page 45
- [debug dot11 mgmt msg](#), on page 46
- [debug dot11 mgmt ssid](#), on page 47
- [debug dot11 mgmt state-machine](#), on page 48
- [debug dot11 mgmt station](#), on page 49
- [debug dot1x](#), on page 50
- [debug dtls](#), on page 51
- [debug fastpath](#) , on page 52
- [debug flexconnect avc](#), on page 57
- [debug flexconnect aaa](#), on page 58
- [debug flexconnect acl](#), on page 59
- [debug flexconnect cckm](#), on page 60
- [debug group](#), on page 61
- [debug fmchs](#), on page 62
- [debug flexconnect client ap](#), on page 63
- [debug flexconnect client ap syslog](#), on page 64
- [debug flexconnect client group](#), on page 65
- [debug flexconnect client group syslog](#), on page 66
- [debug flexconnect group](#), on page 67
- [debug ft](#), on page 68
- [debug hotspot](#), on page 69
- [debug ipv6](#), on page 70

# debug aaa

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

```
debug aaa { [all | avp-xml | detail | events | packet | ldap | local-auth | tacacs] [enable | disable] }
```

Syntax Description		
<b>all</b>	(Optional) Configures the debugging of all AAA messages.	
<b>avp-xml</b>	(Optional) Configures debug of AAA Avp xml events.	
<b>detail</b>	(Optional) Configures the debugging of AAA errors.	
<b>events</b>	(Optional) Configures the debugging of AAA events.	
<b>packet</b>	(Optional) Configures the debugging of AAA packets.	
<b>ldap</b>	(Optional) Configures the debugging of the AAA Lightweight Directory Access Protocol (LDAP) events.	
<b>local-auth</b>	(Optional) Configures the debugging of the AAA local Extensible Authentication Protocol (EAP) events.	
<b>tacacs</b>	(Optional) Configures the debugging of the AAA TACACS+ events.	
<b>enable</b>	(Optional) Enables the debugging.	
<b>disable</b>	(Optional) Disables the 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 AAA LDAP events:

```
(Cisco Controller) > debug aaa ldap enable
```

**Related Commands**

- debug aaa local-auth eap**
- show running-config**

## debug aaa events

To configure the debugging related to DNS-based ACLs, use the **debug aaa events enable** command.

**debug aaa events enable**

---

<b>Syntax Description</b>	<b>events</b> Configures the debugging of DNS-based ACLs.
---------------------------	---

---

---

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	7.6	This command is introduced.

---

The following example shows how to enable the debugging for DNS-based ACLs:

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

# debug aaa local-auth

To configure the debugging of AAA local authentication on the controller, use the **debug aaa local-auth** command.

```
debug aaa local-auth { db | shim | eap { framework | method } { all | errors |
events | packets | sm }} { enable | disable }
```

Syntax Description		
	<b>db</b>	Configures the debugging of the AAA local authentication back-end messages and events.
	<b>shim</b>	Configures the debugging of the AAA local authentication shim layer events.
	<b>eap</b>	Configures the debugging of the AAA local Extensible Authentication Protocol (EAP) authentication.
	<b>framework</b>	Configures the debugging of the local EAP framework.
	<b>method</b>	Configures the debugging of local EAP methods.
	<b>all</b>	Configures the debugging of local EAP messages.
	<b>errors</b>	Configures the debugging of local EAP errors.
	<b>events</b>	Configures the debugging of local EAP events.
	<b>packets</b>	Configures the debugging of local EAP packets.
	<b>sm</b>	Configures the debugging of the local EAP state machine.
	<b>enable</b>	Starts the debugging.
	<b>disable</b>	Stops the 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 the AAA local EAP authentication:

```
(Cisco Controller) > debug aaa local-auth eap method all enable
```

**Related Commands**

- `clear stats local-auth`
- `config local-auth active-timeout`

```
config local-auth eap-profile
config local-auth method fast
config local-auth user-credentials
show local-auth certificates
show local-auth config
show local-auth statistics
```

## debug airewave-director

To configure the debugging of Airewave Director software, use the **debug airewave-director** command.

**debug airewave-director** {**all** | **channel** | **detail** | **error** | **group** | **manager** | **message** | **packet** | **power** | **profile** | **radar** | **rf-change**} {**enable** | **disable**}

Syntax	Description
<b>all</b>	Configures the debugging of all Airewave Director logs.
<b>channel</b>	Configures the debugging of the Airewave Director channel assignment protocol.
<b>detail</b>	Configures the debugging of the Airewave Director detail logs.
<b>error</b>	Configures the debugging of the Airewave Director error logs.
<b>group</b>	Configures the debugging of the Airewave Director grouping protocol.
<b>manager</b>	Configures the debugging of the Airewave Director manager.
<b>message</b>	Configures the debugging of the Airewave Director messages.
<b>packet</b>	Configures the debugging of the Airewave Director packets.
<b>power</b>	Configures the debugging of the Airewave Director power assignment protocol and coverage hole detection.
<b>profile</b>	Configures the debugging of the Airewave Director profile events.
<b>radar</b>	Configures the debugging of the Airewave Director radar detection/avoidance protocol.
<b>rf-change</b>	Configures the debugging of the Airewave Director rf changes.
<b>enable</b>	Enables the Airewave Director debugging.
<b>disable</b>	Disables the Airewave Director debugging.
<b>Command Default</b>	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 Airewave Director profile events:

```
(Cisco Controller) > debug airewave-director profile enable
```

**Related Commands**

- debug disable-all
- show sysinfo



# debug ap

To configure the remote debugging of Cisco lightweight access points or to remotely execute a command on a lightweight access point, use the **debug ap** command.

**debug ap** { **enable** | **disable** | **command** *cmd* } *cisco\_ap*

Syntax Description		
<b>enable</b>	Enables the debugging on a lightweight access point.	<p><b>Note</b> The debugging information is displayed only to the controller console and does not send output to a controller Telnet/SSH CLI session.</p>
<b>disable</b>	Disables the debugging on a lightweight access point.	<p><b>Note</b> The debugging information is displayed only to the controller console and does not send output to a controller Telnet/SSH CLI session.</p>
<b>command</b>	Specifies that a CLI command is to be executed on the access point.	
<i>cmd</i>	Command to be executed.	<p><b>Note</b> The command to be executed must be enclosed in double quotes, such as <b>debug ap command "led flash 30" AP03</b>.</p> <p>The output of the command displays only to the controller console and does not send output to a controller Telnet/SSH CLI session.</p>
<i>cisco_ap</i>	Name of a Cisco lightweight access point.	

**Command Default** The remote debugging of Cisco lightweight access points 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 remote debugging on access point AP01:

```
(Cisco Controller) >debug ap enable AP01
```

The following example shows how to execute the **config ap location** command on access point AP02:

```
(Cisco Controller) >debug ap command "config ap location "Building 1" AP02"
```

The following example shows how to execute the flash LED command on access point AP03:

```
(Cisco Controller) >debug ap command "led flash 30" AP03
```

# debug ap enable

To configure the remote debugging of Cisco lightweight access points or to remotely execute a command on a lightweight access point, use the **debug ap enable** command.

**debug ap** {**enable** | **disable** | **command cmd**} *cisco\_ap*

Syntax Description	enable	Description
		Enables the remote debugging.
	<b>Note</b>	The debugging information is displayed only to the controller console and does not send output to a controller Telnet/SSH CLI session.
	<b>disable</b>	Disables the remote debugging.
	<b>command</b>	Specifies that a CLI command is to be executed on the access point.
	<i>cmd</i>	Command to be executed.
	<b>Note</b>	The command to be executed must be enclosed in double quotes, such as <b>debug ap command "led flash 30" AP03</b> .  The output of the command displays only to the controller console and does not send output to a controller Telnet/SSH CLI session.
	<i>cisco_ap</i>	Cisco lightweight access point name.
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 remote debugging on access point AP01:

```
(Cisco Controller) >debug ap enable AP01
```

The following example shows how to disable the remote debugging on access point AP02:

```
(Cisco Controller) >debug ap disable AP02
```

The following example shows how to execute the flash LED command on access point AP03:

```
(Cisco Controller) >debug ap command "led flash 30" AP03
```

## debug ap packet-dump

To configure the debugging of Packet Capture, use the **debug ap packet-dump** command.

```
debug ap packet-dump { enable | disable }
```

### Syntax Description

**enable** Enables the debugging of Packet Capture of an access point.

**disable** Disables the debugging of Packet Capture of an access point.

### Command Default

Debugging of Packet Capture is disabled.

### Command History

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

### Usage Guidelines

Packet Capture does not work during inter-controller roaming.

The controller does not capture packets created in the radio firmware and sent out of the access point, such as beacon or probe response. Only packets that flow through the radio driver in the Tx path will be captured.

The following example shows how to enable the debugging of Packet Capture from an access point:

```
(Cisco Controller) >debug ap packet-dump enable
```

## debug ap show stats

To debug video messages and statistics of Cisco lightweight access points, use the **debug ap show stats** command.

```
debug ap show stats {802.11a | 802.11b} cisco_ap {tx-queue | packet | load | multicast | client {client_MAC | video | all} | video metrics}
```

```
debug ap show stats video cisco_ap {multicast mgid mgid_database_number | admission | bandwidth}
```

Syntax	Description
<b>802.11a</b>	Specifies the 802.11a network.
<b>802.11b</b>	Specifies the 802.11b/g network.
<i>cisco_ap</i>	Cisco lightweight access point name.
<b>tx-queue</b>	Displays the transmit queue traffic statistics of the AP.
<b>packet</b>	Displays the packet statistics of the AP.
<b>load</b>	Displays the QoS Basic Service Set (QBSS) and other statistics of the AP.
<b>multicast</b>	Displays the multicast supported rate statistics of the AP.
<b>client</b>	Displays the specified client metric statistics.
<i>client_MAC</i>	MAC address of the client.
<b>video</b>	Displays video statistics of all clients on the AP.
<b>all</b>	Displays statistics of all clients on the AP.
<b>video metrics</b>	Displays the video metric statistics.
<b>mgid</b>	Displays detailed multicast information for a single multicast group ID (MGID).
<i>mgid_database_number</i>	Layer 2 MGID database number.
<b>admission</b>	Displays video admission control on the AP.
<b>bandwidth</b>	Displays video bandwidth on the AP.
<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 troubleshoot the access point AP01's transmit queue traffic on an 802.11a network:

```
(Cisco Controller) >debug ap show stats 802.11a AP01 tx-queue
```

The following example shows how to troubleshoot the access point AP02's multicast supported rates on an 802.11b/g network:

```
(Cisco Controller) >debug ap show stats 802.11b AP02 multicast
```

The following example shows how to troubleshoot the metrics of a client identified by its MAC address, associated with the access point AP01 on an 802.11a network:

```
(Cisco Controller) >debug ap show stats 802.11a AP01 client 00:40:96:a8:f7:98
```

The following example shows how to troubleshoot the metrics of all clients associated with the access point AP01 on an 802.11a network:

```
(Cisco Controller) >debug ap show stats 802.11a AP01 client all
```

# debug ap show stats video

To configure the debugging of video messages and statistics of Cisco lightweight access points, use the **debug ap show stats video** command.

```
debug ap show stats video cisco_ap { multicast mgid mgid_value | admission | bandwidth }
```

Syntax Description		
<i>cisco_ap</i>		Cisco lightweight access point name.
<b>multicast mgid</b>		Displays multicast database related information for the specified MGID of an access point.
<i>mgid_value</i>		Layer 2 MGID database number from 1 to 4095.
<b>admission</b>		Displays the video admission control.
<b>bandwidth</b>		Displays the video bandwidth.

**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 an access point AP01's multicast group that is identified by the group's Layer 2 MGID database number:

```
(Cisco Controller) >debug ap show stats video AP01 multicast mgid 50
```

This example shows how to configure the debugging of an access point AP01's video bandwidth:

```
(Cisco Controller) >debug ap show stats video AP01 bandwidth
```

# debug arp

To configure the debugging of Address Resolution Protocol (ARP) options, use the **debug arp** command.

**debug arp** {**all** | **detail** | **events** | **message**} {**enable** | **disable**}

## Syntax Description

<b>all</b>	Configures the debugging of all ARP logs.
<b>detail</b>	Configures the debugging of ARP detail messages.
<b>error</b>	Configures the debugging of ARP errors.
<b>message</b>	Configures the debugging of ARP messages.
<b>enable</b>	Enables the ARP debugging.
<b>disable</b>	Disables the ARP 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 ARP debug settings:

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

The following example shows how to disable ARP debug settings:

```
(Cisco Controller) > debug arp error disable
```

## Related Commands

**debug disable-all**

**show sysinfo**



# debug avc

To configure the debugging of Application Visibility and Control (AVC) options, use the **debug avc error** command.

```
debug avc {events | error} {enable | disable}
```

## Syntax Description

<b>events</b>	Configures the debugging of AVC events.
<b>error</b>	Configures the debugging of AVC errors.
<b>enable</b>	Enables the debugging of AVC events or errors.
<b>disable</b>	Disables the debugging of AVC events or errors.

## Command Default

By default, the debugging of AVC options 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 AVC errors:

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

## Related Commands

- config avc profile delete**
- config avc profile rule**
- config wlan avc**
- show avc profile**
- show avc applications**
- show avc statistics**

# debug bcast

To configure the debugging of broadcast options, use the **debug bcast** command.

**debug bcast** {**all** | **error** | **message** | **igmp** | **detail**} {**enable** | **disable**}

Syntax Description		
	<b>all</b>	Configures the debugging of all broadcast logs.
	<b>error</b>	Configures the debugging of broadcast errors.
	<b>message</b>	Configures the debugging of broadcast messages.
	<b>igmp</b>	Configures the debugging of broadcast IGMP messages.
	<b>detail</b>	Configures the debugging of broadcast detailed messages.
	<b>enable</b>	Enables the broadcast debugging.
	<b>disable</b>	Disables the broadcast 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 broadcast messages:

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

The following example shows how to disable the debugging of broadcast messages:

```
(Cisco Controller) > debug bcast message disable
```

**Related Commands**

- debug disable-all**
- show sysinfo**

# debug call-control

To configure the debugging of the SIP call control settings, use the **debug call-control** command.

**debug call-control** {all | event} {enable | disable}

<b>Syntax Description</b>	<b>all</b>	Configures the debugging options for all SIP call control messages.
	<b>event</b>	Configures the debugging options for SIP call control events.
	<b>enable</b>	Enables the debugging of SIP call control messages or events.
	<b>disable</b>	Disables the debugging of SIP call control messages or events.
<b>Command Default</b>	Disabled.	
<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 the debugging of all SIP call control messages:

```
(Cisco Controller) >debug call-control all enable
```

# debug capwap

To configure the debugging of Control and Provisioning of Wireless Access Points (CAPWAP) settings, use the **debug capwap** command.

```
debug capwap {detail | dtls-keepalive | errors | events | hexdump | info | packet | payload | mfp} {enable | disable}
```

## Syntax Description

<b>detail</b>	Configures the debugging for CAPWAP detail settings.
<b>dtls-keepalive</b>	Configures the debugging for CAPWAP DTLS data keepalive packets settings.
<b>errors</b>	Configures the debugging for CAPWAP error settings.
<b>events</b>	Configures the debugging for CAPWAP events settings.
<b>hexdump</b>	Configures the debugging for CAPWAP hexadecimal dump settings.
<b>info</b>	Configures the debugging for CAPWAP info settings.
<b>packet</b>	Configures the debugging for CAPWAP packet settings.
<b>payload</b>	Configures the debugging for CAPWAP payload settings.
<b>mfp</b>	Configures the debugging for CAPWAP mfp settings.
<b>enable</b>	Enables the debugging of the CAPWAP command.
<b>disable</b>	Disables the debugging of the CAPWAP command.

## 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 CAPWAP details:

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

## debug capwap reap

To configure the debugging of Control and Provisioning of Wireless Access Points (CAPWAP) settings on a FlexConnect access point, use the **debug capwap reap** command.

**debug capwap reap** [**mgmt** | **load**]

<b>Syntax Description</b>	<b>mgmt</b>	(Optional) Configures the debugging for client authentication and association messages.
	<b>load</b>	(Optional) Configures the debugging for payload activities, which is useful when the FlexConnect access point boots up in standalone mode.
<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 configure the debugging of FlexConnect client authentication and association messages:

```
(Cisco Controller) >debug capwap reap mgmt
```

## debug ccxdia

To configure debugging of Cisco Compatible Extensions (CCX) diagnostic options, use the **debug ccxdia** command.

**debug ccxdia** {all | error | event | packet} {enable | disable}

### Syntax Description

<b>all</b>	Configures debugging of all the CCX S69 messages.
<b>error</b>	Configures debugging of the CCX S69 errors.
<b>event</b>	Configures debugging of the CCX S69 events.
<b>packet</b>	Configures debugging of the CCX S69 packets.
<b>enable</b>	Enables debugging of the CCX S69 options.
<b>disable</b>	Disables debugging of the CCX S69 options.

### 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 CCX S69 packets debugging:

```
(Cisco Controller) >debug ccxdia packets enable
```

## debug ccxrm

To configure debugging of the CCX Cisco Client eXtension (CCX) Radio Management (RM), use the **debug ccxrm** command.

```
debug ccxrm { all | detail | error | location-calibration | message | packet | warning }
{ enable | disable }
```

Syntax Description		
	<b>all</b>	Configures debugging of all CCX RM messages.
	<b>detail</b>	Configures detailed debugging of CCX RM.
	<b>error</b>	Configures debugging of the CCX RM errors.
	<b>location-calibration</b>	Configures debugging of the CCX RM location calibration.
	<b>message</b>	Configures debugging of CCX RM messages.
	<b>packet</b>	Configures debugging of the CCX RM packets.
	<b>warning</b>	Configures debugging of the CCX RM warnings.
	<b>enable</b>	Enables debugging of the CCX RM options.
	<b>disable</b>	Disables debugging of the CCX RM options.

**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 CCX RM debugging:

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

## debug ccxs69

To configure debugging of CCX S69 tasks, use the **debug ccxs69** command.

```
debug ccxs69 {all | error | event} {enable | disable}
```

Syntax Description		
<b>all</b>	Configures debugging of all the CCX S69 messages.	
<b>error</b>	Configures debugging of the CCX S69 errors.	
<b>event</b>	Configures debugging of the CCX S69 events.	
<b>enable</b>	Enables debugging of the CCX S69 options.	
<b>disable</b>	Disables debugging of the CCX S69 options.	

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 CCX S69 debugging:

```
(Cisco Controller) >debug ccxs69 all enable
```



# debug cckm

To configure the debugging of the Cisco Centralized Key Management options, use the **debug cckm**

```
debug cckm {client | detailed} {enable | disable}
```

## Syntax Description

<b>client</b>	Configures debugging of the Cisco Centralized Key Management of clients.
<b>detailed</b>	Configures detailed debugging of Cisco Centralized Key Management.
<b>enable</b>	Enables debugging of Cisco Centralized Key Management.
<b>disable</b>	Disables debugging of Cisco Centralized Key Management.

## 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 detailed debugging of Cisco Centralized Key Management:

```
(Cisco Controller) > debug cckm detailed enable
```

# debug client

To configure the debugging for a specific client, use the **debug client** command.

**debug client** *mac\_address*

<b>Syntax Description</b>	<i>mac_address</i>	MAC address of the client.
<b>Command Default</b>	None	
<b>Usage Guidelines</b>	After entering the <b>debug client</b> <i>mac_address</i> command, if you enter the <b>debug aaa events enable</b> command, then the AAA events logs are displayed for that particular client MAC address.	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	7.6	This command was introduced.

The following example shows how to debug a specific client:

```
(Cisco Controller) > debug client 01:35:6x:yy:21:00
```

## debug cts aaa

To configure the Cisco TrustSec AAA debug options, use the **debug cts aaa** command.

```
debug cts aaa {all | errors | events} {enable | disable}
```

---

**Syntax Description**

---

**all** Configures debugging of all the CTS AAA debug options

---

**errors** Configures debugging of all the CTS AAA errors

---

**events** Configures debugging of all the CTS AAA events

---

**enable** Enables debugging

---

**disable** Disables debugging

---

---

**Command Default**

None

---

**Command History**

---

**Release**

---

**Modification**

---

8.4

---

This command was introduced.

---

## debug cts authz

To configure the Cisco TrustSec security group access control list (SGACL) download debug options, use the **debug cts authz** command.

**debug cts authz** { **aaa** | **all** | **errors** | **events** } { **enable** | **disable** }

### Syntax Description

<b>aaa</b>	Configures debugging of CTS AAA policy
<b>all</b>	Configures debugging of all the CTS policies
<b>errors</b>	Configures debugging of all the CTS policy errors
<b>events</b>	Configures debugging of all the CTS policy events
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

### Command Default

None

### Command History

Release	Modification
8.4	This command was introduced.

## debug cts capwap

To configure the debug options for Cisco TrustSec policy download over CAPWAP messages, use the **debug cts capwap** command.

**debug cts capwap** {**messages** | **all** | **errors** | **events**} {**enable** | **disable**}

Syntax Description	
<b>messages</b>	Configures debugging of Protected Access Credential (PAC) CAPWAP messages
<b>all</b>	Configures debugging of all the CTS CAPWAP messages
<b>errors</b>	Configures debugging of the PAC CAPWAP errors
<b>events</b>	Configures debugging of the PAC CAPWAP events
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

Command Default	None
-----------------	------

Command History	Release	Modification
	8.4	This command was introduced.

## debug cts env-data

To configure Cisco TrustSec environment data debugs, use the **debug cts env-data** command.

**debug cts env-data** { **all** | **errors** | **events** } { **enable** | **disable** }

### Syntax Description

<b>all</b>	Configures debugging of all the CTS environment data
<b>errors</b>	Configures debugging of CTS environment data errors
<b>events</b>	Configures debugging of CTS environment data events
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

### Command Default

None

### Command History

Release	Modification
8.4	This command was introduced.

## debug cts ha

To configure the Cisco TrustSec High Availability (HA) debug options, use the **debug cts ha** command.

```
debug cts ha {all | errors | events} {enable | disable}
```

---

**Syntax Description**

<b>all</b>	Configures debugging of all the CTS HA options
<b>errors</b>	Configures debugging of CTS HA errors
<b>events</b>	Configures debugging of CTS HA events
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

---

**Command Default**

None

---

**Command History**

<b>Release</b>	<b>Modification</b>
8.4	This command was introduced.

## debug cts key-store

To configure the Cisco TrustSec Key-store debug options, use the **debug cts key-store** command.

**debug cts key-store** { **enable** | **disable** }

---

### Syntax Description

**enable** Enables debugging

**disable** Disables debugging

---



---

### Command Default

None

---

### Command History

---

#### Release

8.4

---

#### Modification

This command was introduced.

---



## debug cts provisioning

To configure the Cisco TrustSec PAC Provisioning debug options, use the **debug cts provisioning** command.

```
debug cts provisioning {packets | all | errors | events} {enable | disable}
```

### Syntax Description

<b>packets</b>	Configures debugging of PAC provisioning packets
<b>all</b>	Configures debugging of all the PAC provisioning options
<b>errors</b>	Configures debugging of the PAC provisioning errors
<b>events</b>	Configures debugging of the PAC provisioning events
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

### Command Default

None

### Command History

Release	Modification
8.4	This command was introduced.

## debug cts sgt

To configure debugging of up to 10 SGTs, use the **debug cts sgt** command.

**debug cts sgt** {*sgt-1* | *sgt-2* | *sgt-3* | *sgt-4* | *sgt-5* | *sgt-6* | *sgt-7* | *sgt-8* | *sgt-9* | *sgt-10*}

---

### Syntax Description

*sgt-1* to  
*sgt-10* SGT IDs that you have to enter.

---



---

### Command Default

None

---



---

### Command History

Release	Modification
8.4	This command was introduced.

---

## debug cts sxp

To configure debugging of Cisco TrustSec SXP options, use the **debug cts sxp** command.

**debug cts sxp** { **all** | **errors** | **events** | **framework** | **message** } { **enable** | **disable** }

### Syntax Description

<b>all</b>	Configures debugging of all the CTS SXP options
<b>errors</b>	Configures debugging of the CTS SXP errors
<b>events</b>	Configures debugging of the CTS SXP events
<b>framework</b>	Configures debugging of the CTS SXP framework
<b>message</b>	Configures debugging of the CTS SXP messages
<b>enable</b>	Enables debugging
<b>disable</b>	Disables debugging

### Command Default

None

### Command History

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

# debug cac

To configure the debugging of Call Admission Control (CAC) options, use the **debug cac** command.

```
debug cac {all | event | packet} {enable | disable}
```

## Syntax Description

<b>all</b>	Configures the debugging options for all CAC messages.
<b>event</b>	Configures the debugging options for CAC events.
<b>packet</b>	Configures the debugging options for selected CAC packets.
<b>kts</b>	Configures the debugging options for KTS-based CAC messages.
<b>enable</b>	Enables the debugging of CAC settings.
<b>disable</b>	Disables the debugging of CAC settings.

## Command Default

By default, the debugging of CAC options 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 debugging of CAC settings:

```
(Cisco Controller) > debug cac event enable
(Cisco Controller) > debug cac packet enable
```

## Related Commands

```
config 802.11 cac video acm
config 802.11 cac video max-bandwidth
config 802.11 video roam-bandwidth
config 802.11 cac video tspec-inactivity-timeout
config 802.11 cac voice load-based
config 802.11 cac voice roam-bandwidth
config 802.11 cac voice stream-size
config 802.11 cac voice tspec-inactivity-timeout
```

# debug cdp

To configure debugging of CDP, use the **debug cdp** command.

```
debug cdp {events | packets} {enable | disable}
```

---

**Syntax Description**

---

**events** Configures debugging of the CDP events.

---

**packets** Configures debugging of the CDP packets.

---

**enable** Enables debugging of the CDP options.

---

**disable** Disables debugging of the CDP options.

---

---

**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 CDP event debugging in a Cisco controller:

```
(Cisco Controller) > debug cdp
```

# debug crypto

To configure the debugging of the hardware cryptographic options, use the **debug crypto** command.

**debug crypto** {all | sessions | trace | warning} {enable | disable}

## Syntax Description

<b>all</b>	Configures the debugging of all hardware crypto messages.
<b>sessions</b>	Configures the debugging of hardware crypto sessions.
<b>trace</b>	Configures the debugging of hardware crypto sessions.
<b>warning</b>	Configures the debugging of hardware crypto sessions.
<b>enable</b>	Enables the debugging of hardware cryptographic sessions.
<b>disable</b>	Disables the debugging of hardware cryptographic sessions.

## 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 hardware crypto sessions:

```
(Cisco Controller) > debug crypto sessions enable
```

## Related Commands

**debug disable-all**  
**show sysinfo**

# debug dhcp

To configure the debugging of DHCP, use the **debug dhcp** command.

```
debug dhcp {message | packet} {enable | disable}
```

## Syntax Description

<b>message</b>	Configures the debugging of DHCP error messages.
<b>packet</b>	Configures the debugging of DHCP packets.
<b>enable</b>	Enables the debugging DHCP messages or packets.
<b>disable</b>	Disables the debugging of DHCP messages or packets.

## Command Default

None

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

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

## debug dhcp service-port

To enable or disable debugging of the Dynamic Host Configuration Protocol (DHCP) packets on the service port, use the **debug dhcp service-port** command.

**debug dhcp service-port** { **enable** | **disable** }

Syntax Description		
	<b>enable</b>	Enables the debugging of DHCP packets on the service port.
	<b>disable</b>	Disables the debugging of DHCP packets on the service port.
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 DHCP packets on a service port:

```
(Cisco Controller) >debug dhcp service-port enable
```



# debug disable-all

To disable all debug messages, use the **debug disable-all** command.

## **debug disable-all**

---

**Syntax Description**

This command has no arguments or keywords.

---

**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 disable all debug messages:

```
(Cisco Controller) > debug disable-all
```

# debug dns

To configure debugging of Domain Name System (DNS) options, use the **debug dns** command.

**debug dns** {**all** | **detail** | **error** | **message**} {**enable** | **disable**}

## Syntax Description

<b>all</b>	Configures debugging of all the DNS options.
<b>detail</b>	Configures debugging of the DNS details.
<b>error</b>	Configures debugging of the DNS errors.
<b>message</b>	Configures debugging of the DNS messages.
<b>enable</b>	Enables debugging of the DNS options.
<b>disable</b>	Disables debugging of the DNS options.

## 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 DNS error debugging:

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

# debug dot11

To configure the debugging of 802.11 events, use the **debug dot11** command.

```
debug dot11 {all | load-balancing | management | mobile | nmsp | probe | rldp | rogue
| state} {enable | disable}
```

Syntax Description		
<b>all</b>		Configures the debugging of all 802.11 messages.
<b>load-balancing</b>		Configures the debugging of 802.11 load balancing events.
<b>management</b>		Configures the debugging of 802.11 MAC management messages.
<b>mobile</b>		Configures the debugging of 802.11 mobile events.
<b>nmsp</b>		Configures the debugging of the 802.11 NMSP interface events.
<b>probe</b>		Configures the debugging of probe.
<b>rldp</b>		Configures the debugging of 802.11 Rogue Location Discovery.
<b>rogue</b>		Configures the debugging of 802.11 rogue events.
<b>state</b>		Configures the debugging of 802.11 mobile state transitions.
<b>enable</b>		Enables the 802.11 debugging.
<b>disable</b>		Disables the 802.11 debugging.
<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 the debugging of 802.11 settings:

```
(Cisco Controller) > debug dot11 state enable
(Cisco Controller) > debug dot11 mobile enable
```

# debug dot11

To configure the debugging of 802.11 events, use the **debug dot11** command.

```
debug dot11 {all | load-balancing | management | mobile | nmosp | probe | rldp | rogue
| state} {enable | disable}
```

## Syntax Description

<b>all</b>	Configures the debugging of all 802.11 messages.
<b>load-balancing</b>	Configures the debugging of 802.11 load balancing events.
<b>management</b>	Configures the debugging of 802.11 MAC management messages.
<b>mobile</b>	Configures the debugging of 802.11 mobile events.
<b>nmosp</b>	Configures the debugging of the 802.11 NMSP interface events.
<b>probe</b>	Configures the debugging of probe.
<b>rldp</b>	Configures the debugging of 802.11 Rogue Location Discovery.
<b>rogue</b>	Configures the debugging of 802.11 rogue events.
<b>state</b>	Configures the debugging of 802.11 mobile state transitions.
<b>enable</b>	Enables the 802.11 debugging.
<b>disable</b>	Disables the 802.11 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 802.11 settings:

```
(Cisco Controller) > debug dot11 state enable
(Cisco Controller) > debug dot11 mobile enable
```

# debug dot11 mgmt interface

To configure debugging of 802.11 management interface events, use the **debug dot11 mgmt interface** command.

## **debug dot11 mgmt interface**

---

<b>Syntax Description</b>	This command has no arguments or keywords.
---------------------------	--

---

<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 debug 802.11 management interface events:

```
(Cisco Controller) >debug dot11 mgmt interface
```

## debug dot11 mgmt msg

To configure debugging of 802.11 management messages, use the **debug dot11 mgmt msg** command.

### **debug dot11 mgmt msg**

---

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

---

**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.

---

This example shows how to debug dot11 management messages:

```
(Cisco Controller) >debug dot11 mgmt msg
```

# debug dot11 mgmt ssid

To configure debugging of 802.11 SSID management events, use the **debug dot11 mgmt ssid** command.

**debug dot11 mgmt ssid**

---

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

---

**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 802.11 SSID management events:

```
(Cisco Controller) >debug dot11 mgmt ssid
```

## debug dot11 mgmt state-machine

To configure debugging of the 802.11 state machine, use the **debug dot11 mgmt state-machine** command.

### **debug dot11 mgmt state-machine**

---

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

---

**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 802.11 state machine:

```
(Cisco Controller) >debug dot11 mgmt state-machine
```



# debug dot11 mgmt station

To configure the debugging of the management station settings, use the **debug dot11 mgmt station** command.

## **debug dot11 mgmt station**

---

**Syntax Description**

This command has no arguments or keywords.

---

**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.

The following example shows how to configure the debugging of the management station settings:

```
(Cisco Controller) >debug dot11 mgmt station
```

# debug dot1x

To configure debugging of the 802.1X options, use the **debug dot1x** command.

**debug dot1x** {**aaa** | **all** | **events** | **packets** | **states**} {**enable** | **disable**}

## Syntax Description

<b>aaa</b>	Configures debugging of the 802.1X AAA interactions.
<b>all</b>	Configures debugging of all the 802.1X messages.
<b>events</b>	Configures debugging of the 802.1X events.
<b>packets</b>	Configures debugging of the 802.1X packets.
<b>states</b>	Configures debugging of the 802.1X state transitions.
<b>enable</b>	Enables debugging of the 802.1X options.
<b>disable</b>	Disables debugging of the 802.1X options.

## 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 802.1X state transitions debugging:

```
(Cisco Controller) > debug dot1x states enable
```

# debug dtls

To configure debugging of the Datagram Transport Layer Security (DTLS) options, use the **debug dtls** command.

**debug dtls** {all | event | packet | trace} {enable | disable}

## Syntax Description

<b>all</b>	Configures debugging of all the DTLS messages.
<b>event</b>	Configures debugging of the DTLS events.
<b>packet</b>	Configures debugging of the DTLS packets.
<b>trace</b>	Configures debugging of the DTLS trace messages.
<b>enable</b>	Enables debugging of the DTLS options.
<b>disable</b>	Disables debugging of the DTLS options.

## Command Default

None

## Command History

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

## Usage Guidelines

The debug actions described here are used in conjunction with CAPWAP troubleshooting.

The following example shows how to enable DTLS packet debugging:

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

## debug fastpath

To debug the issues in the 10-Gigabit Ethernet interface of the controller and to view details of all the management and control features of the controller, use the **debug fastpath** command.

```
debug fastpath [{disable | enable | errors | events | warning | log | status | dump | audit | clear}]
debug fastpath log [{error events show}]
debug fastpath dump [{stats DP_number} | {fpapoolDP_number} | {ownerdb} | {portdb}
| {preauthurl client-mac | 0 | 1 | All} | {tun4dbindexDP_number} | {scbdbindexDP_number} |
{cfgtool -- dump.sfp} | {urlaclldbstart-acl-id start-rule-index} | {vlandb} | {dpcp-stats} | {clear
stats} | {systemdb} | {debug | {wlanappstatswlan_id}} | {appqosdb}]
```

Syntax	Description
<b>disable</b>	Enables debug of fastpath messages.
<b>enable</b>	Disables debug of fastpath messages.
<b>errors</b>	Displays the debug messages related to the fastpath errors.
<b>events</b>	Displays the debug messages related to the fastpath events.
<b>warnings</b>	Displays the debug messages related to the fastpath warnings.
<b>log</b>	Configures debug of log messages.
<i>errors</i>	Configures debug of fastpath errors.
<i>events</i>	Configures debug of fastpath events.
<i>show</i>	Displays log of most recent events related to fastpath.
<b>status</b>	Displays status of fastpath configuration.
<b>dump</b>	Displays the CLI dump commands.
<b>stats</b>	Displays the debug statistics from the data plane.
<i>DP_number</i>	Displays the statistic counters at data plane based on selected data plane number. Values include 0, 1, and All. The default option is All. You must select: <ul style="list-style-type: none"> <li>• The index 0 for the Cisco Wireless LAN Controller 2504 Series, Cisco Wireless LAN Controller 5508 Series, Cisco Wireless LAN Controller 7500 Series, Cisco Wireless LAN Controller 8500 Series.</li> <li>• The index 0 and/or 1 respectively for the two data planes in WiSM2 to view statistics of individual data plane or from both.</li> </ul>

<b>fpapool</b>	Displays statistics of packet buffer in data plane.
<i>DP_number</i>	Displays statistics of packet buffer based on data plane number. Values include 0, 1, and All. The default option is All. You must select: <ul style="list-style-type: none"> <li>• The index 0 for the Cisco Wireless LAN Controller 2504 Series, Cisco Wireless LAN Controller 5508 Series, Cisco Wireless LAN Controller 7500 Series, Cisco Wireless LAN Controller 8500 Series.</li> <li>• The index 0 and/or 1 respectively for the two data planes in WiSM2 to view statistics of individual data plane or from both.</li> </ul>
<b>ownerdb</b>	Displays the data plane owner information.
<b>portdb</b>	Displays the port database at data plane.
<b>preauthurl</b>	Preauth URL learned IPs.
<i>client-mac</i>	MAC address of the client.
0   1   All	Dataplane options.
<b>tun4db</b>	Dumps the first 20 tunnels from the data plane.
<i>index</i>	Dumps 20 tunnel entries from index provided. You must use data plane number 0/1 to denote WISM2 data plane processor.
<i>DP_number</i>	Dumps the first twenty client entries from the data plane. Values include 0, 1, and All. The default option is All. You must select: <ul style="list-style-type: none"> <li>• The index 0 for the Cisco Wireless LAN Controller 2504 Series, Cisco Wireless LAN Controller 5508 Series, Cisco Wireless LAN Controller 7500 Series, Cisco Wireless LAN Controller 8500 Series.</li> <li>• The index 0 and/or 1 respectively for the two data planes in WiSM2 to view statistics of individual data plane or from both.</li> </ul>
<b>scbdb</b>	Dumps 20 client entries starting from index provided. You must use data plane number 0/1 to denote WISM2 data plane processor.
<i>index</i>	Dumps client information for the selected MAC address.

<i>DP_number</i>	Dumps the first twenty client entries from the data plane. Values include 0, 1, and All. The default option is All. You must select: <ul style="list-style-type: none"> <li>• The index 0 for the Cisco Wireless LAN Controller 2504 Series, Cisco Wireless LAN Controller 5508 Series, Cisco Wireless LAN Controller 7500 Series, Cisco Wireless LAN Controller 8500 Series.</li> <li>• The index 0 and/or 1 respectively for the two data planes in WiSM2 to view statistics of individual data plane or from both.</li> </ul>
<b>cfgtool -- dump.sfp</b>	Displays the model/type of SX/LC/T small form-factor plug-in (SFP) modules with the OUI Partnumber.
<b>urlacldb</b> <i>start-acl-id start-rule-index</i>	Dumps the URL ACL database.
<b>vlandb</b>	Dumps the VLAN database in the dataplane.
<b>dpcp-stats</b>	Displays the dataplane to controlplane message statistics.
<b>clear stats</b>	Clears the data plane statistic counters.
<b>systemdb</b>	Displays the global data plane configuration.
<b>debug</b>	Displays the few latest messages of the data plane to enable troubleshooting.
<b>wlanappstats</b>	Displays Application Visibility and Control (AVC) statistics of a WLAN.
<i>wlan_id</i>	The WLAN identifier of the WLAN you need identify the AVC statistics.
<b>appqosdb</b>	Displays Application Visibility and Control (AVC) database statistics of the data plane.
<b>clear</b>	Clear command.

**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.
	8.3	This command was enhanced in this release. The new keyword added is urlacldb
	8.6	This command was enhanced in this release. The new keyword added is <b>preauthurl</b> .

**Usage Guidelines** None

### Examples

The following is an example of the SX/LC/T small form-factor plug-in (SFP) modules model/type with the respective OUI Partnumber.

```
(Cisco Controller) >debug fastpath status
```

Pr	Type	STP Stat	Admin Mode	Physical Mode	Physical Status	Link Status	Link Trap	POE
1	Normal	Forw	Enable	Auto	1000 Full	Up	Enable	N/A
2	Normal	Forw	Enable	Auto	1000 Full	Up	Enable	N/A

The following is an example of the fastpath status displayed while you execute the status command.

```
(Cisco Controller) >debug fastpath status
```

```
FP0.03:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.00:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.06:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.05:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.06:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.03:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.06:(119115)Received command: FP_CMD_ACL_COUNTER_GET
FP0.07:(119125)Received command: FP_CMD_ACL_COUNTER_GET
FP0.04:(119125)Received command: FP_CMD_ACL_COUNTER_GET
FP0.03:(119125)Received command: FP_CMD_ACL_COUNTER_GET
```

The following is an example of the fastpath errors displayed while you execute the debug fastpath log errors command.

```
(Cisco Controller) >debug fastpath log errors
```

```
FP0.04:(873365) [fp_ingress_capwap:429]Discarding Control/Data
Plane DTLS-Application packets after Lookup Failed
FP0.02:(873418)Change logDebugLevel from: 0x1e to 0x9
```

The following is an example of the fastpath events displayed while you execute the debug fastpath log events command.

```
(Cisco Controller) >debug fastpath log events
```

```
FP0.09:(873796) [fp_ingress_capwap:429]Discarding Control/Dat
a Plane DTLS-Application packets after Lookup Failed
FP0.06:(873921)Change logDebugLevel from: 0x9 to 0x1e
```

The following is an example displayed while you execute the debug fastpath log show command.

```
(Cisco Controller) >debug fastpath log show
```

```
FP0.07:(874033)Change logDebugLevel from: 0x1e to 0x9
Fastpath CPU0.02: FAST CACHE DISABLED
Fastpath CPU0.02: FAST CACHE ENABLED
Fastpath CPU0.00: Received command: FP_CMD_ADD_AP
Fastpath CPU0.05: Received command: FP_CMD_DEL_TUN4 ifTun=1113
```

```
Fastpath CPU0.03: Received command: FP_CMD_DEL_TUN4 ifTun=3161
Fastpath CPU0.03: Received command: FP_CMD_DEL_AP
FP0.02:[cmdDelMcastRgTun:6733]failed to delete mcast rg tun 0 ifTun=3161
FP0.07:[fp_ingress_capwap:429]Discarding Control/Data Plane
DTLS-Application packets after Lookup Failed
FP0.01:[fp_ingress_capwap:429]Discarding Control/Data Plane
DTLS-Application packets after Lookup Failed
Fastpath CPU0.01: Received command: FP_CMD_ADD_TUN4 type=CAPWAP ifTun=1114
dstIP
=9.4.110.100 dstMac=2037.06e2.5ec4 dstIPv6=
0000:0000:0000:0000:0000:0000:0000:0000
Fastpath CPU0.01: Tunnel 1114 srcip=9041820 dstip=9046e64 xor=0x7644(30276)
LAG Offset=0,0,0,0,1,0,1,4
Fastpath CPU0.09: Received command: FP_CMD_ADD_TUN4 type=CAPWAP ifTun=3162
dstIP
=9.4.110.100 dstMac=2037.06e2.5ec4 dstIPv6=
0000:0000:0000:0000:0000:0000:0000:0000
Fastpath CPU0.09: Tunnel 3162 srcip=9041820 dstip=9046e64 xor=0x7644(30276)
LAG Offset=0,0,0,0,1,0,1,4
Fastpath CPU0.00: Received command: FP_CMD_SET_INTERFACE_MTU
Fastpath CPU0.00: FAST CACHE DISABLED
Fastpath CPU0.00: FAST CACHE ENABLED
Fastpath CPU0.00: Received command: FP_CMD_ADD_AP
Fastpath CPU0.03: Received command: FP_CMD_UPDATE_EOIP for index=5122
Fastpath CPU0.02: Received command: FP_CMD_UPDATE_EOIP for index=5122
Fastpath CPU0.00: Received command: FP_CMD_DEL_TUN4 ifTun=1114
Fastpath CPU0.03: Received command: FP_CMD_DEL_TUN4 ifTun=3162
Fastpath CPU0.03: Received command: FP_CMD_DEL_AP
FP0.04:[cmdDelMcastRgTun:6733]failed to delete mcast rg tun 0 ifTun=3162
```



## debug flexconnect avc

To debug a Flexconnect Application Visibility and Control (AVC) event, use the **debug flexconnect avc** command.

**debug flexconnect avc** {**event** | **error** | **detail**} {**enable** | **disable**}

Syntax Description	
<b>event</b>	Debugsa FlexConnect AVC event.
<b>error</b>	Debugs a FlexConnect AVC error.
<b>detail</b>	Debugs a FlexConnect AVC details.
<b>enable</b>	Enables debug.
<b>disable</b>	Disables debug.

Command Default	
	None

Command History	Release	Modification
	8.1	This command was introduced.

The following example shows how to enable a debug action for an event:

```
(Cisco Controller) >debug flexconnect avc event enable
```

## debug flexconnect aaa

To configure debugging of FlexConnect backup RADIUS server events or errors, use the **debug flexconnect aaa** command.

**debug flexconnect aaa** {event | error} {enable | disable}

Syntax Description	event	Configures the debugging for FlexConnect RADIUS server events.
	error	Configures the debugging for FlexConnect RADIUS server errors.
	enable	Enables the debugging of FlexConnect RADIUS server settings.
	disable	Disables the debugging of FlexConnect RADIUS server 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 FlexConnect RADIUS server events:

```
(Cisco Controller) >debug flexconnect aaa event enable
```

# debug flexconnect acl

Configures debugging of FlexConnect access control lists (ACLs), use the **debug flexconnect acl** command.

```
debug flexconnect acl {enable | disable}
```

<b>Syntax Description</b>	<b>enable</b>	Enables the debugging of FlexConnect ACLs.
	<b>disable</b>	Disables the debugging of FlexConnect ACLs.
<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 the debugging of FlexConnect ACLs:

```
(Cisco Controller) >debug flexconnect acl enable
```

## debug flexconnect cckm

Configure debugging of FlexConnect Cisco Centralized Key Management (CCKM) fast roaming, use the **debug flexconnect cckm** command.

**debug flexconnect cckm** {enable | disable}

### Syntax Description

<b>enable</b>	Enables the debugging of FlexConnect CCKM fast roaming settings.
<b>disable</b>	Disables the debugging of FlexConnect CCKM fast roaming 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 FlexConnect CCKM fast roaming events:

```
(Cisco Controller) >debug flexconnect cckm event enable
```

# debug group

To configure the debugging of access point groups, use the **debug group** command.

```
debug group {enable | disable}
```

<b>Syntax Description</b>	<b>enable</b>	Enables the debugging of access point groups.
	<b>disable</b>	Disables the debugging of access point groups.
<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 the debugging of access point groups:

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

## debug fmchs

To configure debugging of Fixed Mobile Convergence Handover Service (FMCHS) of the controller, use the **debug fmchs** command.

**debug fmchs** { **all** | **error** | **event** | **nmsp** | **packet** } { **enable** | **disable** }

### Syntax Description

<b>all</b>	Configures debugging of all FMCHS messages.
<b>error</b>	Configures debugging of the FMCHS errors.
<b>event</b>	Configures debugging of the FMCHS events.
<b>nmsp</b>	Configures debugging of the FMCHS NMSP events.
<b>packet</b>	Configures debugging of the FMCHS packets.
<b>enable</b>	Enables debugging of the FMCHS options.
<b>disable</b>	Disables debugging of the FMCHS options.

### 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 FMCHS event debugging:

```
(Cisco Controller) >debug fmchs event enable
```

## debug flexconnect client ap

To debug FlexConnect client access point MAC addresses, use the **debug flexconnect client ap** command.

```
debug flexconnect client ap ap-name {add | delete} MAC-address1 MAC-address2 MAC-address3
MAC-address4
```

Syntax Description	add	delete
	Adds the MAC address to the group.	Deletes the MAC address from the group.
	<i>MAC-address</i> MAC address of the client	

Command Default	None
-----------------	------

Command History	Release	Modification
	8.1	This command was added

The following example shows how to debug FlexConnect client ap 'room' MAC addresses:

```
(Cisco Controller) >debug flexconnect client ap room add 00.0c.41.07.33.a6 0A.0c.52.17.97.b6
```

## debug flexconnect client ap syslog

To configure debug logging of the syslog server for a FlexConnect client AP, use the **debug flexconnect client ap** command.

```
debug flexconnect client ap ap-name syslog {ip-address | disable}
```

Syntax Description	
	<i>ip-address</i> Configures the syslog server ip-address for debug logging.
	<b>disable</b> Disables the debug logging to the syslog server.

Command Default	
	None

Command History	Release	Modification
	8.1	This command was added.

The following example shows how to configure syslog server for debug log for the FlexConnect client AP 'room':

```
(Cisco Controller) >debug flexconnect client ap room syslog 192.168.1.1
```



## debug flexconnect client group

To debug FlexConnect client group MAC addresses, use the **debug flexconnect client group** command.

**debug flexconnect client group** *group-name* { **add** | **delete** } *MAC-address1* *MAC-address2* *MAC-address3* *MAC-address4*

Syntax Description	add	delete
	Adds the MAC address to the group.	Deletes the MAC address from the group.
	<i>MAC-address</i> MAC address of the client.	
Command Default	None	
Command History	Release	Modification
	8.1	This command was added.

The following example shows how to debug FlexConnect client group MAC addresses:

```
(Cisco Controller) >debug flexconnect client group school add 00.0c.41.07.33.a6
0A.0c.52.17.97.b6
```

## debug flexconnect client group syslog

To debug FlexConnect group access point syslog, use the **debug flexconnect client group** command.

**debug flexconnect client group** *group-name* **syslog** *ip-address* | *disable*

Syntax Description	
<b>ip-address</b>	Configures the syslog server ip-address for debug logging.
<b>disable</b>	Disables the debug logging to the syslog server.

Command Default	
	None

Command History	Release	Modification
	8.1	This command was added.

The following example shows how to configure FlexConnect client group 'school' for debug logging purposes:

```
(Cisco Controller) >debug flexconnect client group school syslog 192.168.1.1
```

## debug flexconnect group

To configure debugging of FlexConnect access point groups, use the **debug flexconnect group** command.

```
debug flexconnect group {enable | disable}
```

<b>Syntax Description</b>	<b>enable</b>	Enables the debugging of FlexConnect access point groups.
	<b>disable</b>	Disables the debugging of FlexConnect access point groups.
<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 the debugging of FlexConnect access point groups:

```
(Cisco Controller) >debug flexconnect group enable
```

# debug ft

To configure debugging of 802.11r, use the **debug ft** command.

**debug ft** { **events** | **keys** } { **enable** | **disable** }

## Syntax Description

<b>events</b>	Configures debugging of the 802.11r events.
<b>keys</b>	Configures debugging of the 802.11r keys.
<b>enable</b>	Enables debugging of the 802.11r options.
<b>disable</b>	Disables debugging of the 802.11r options.

## 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 802.11r debugging:

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

## debug hotspot

To configure debugging of HotSpot events or packets, use the **debug hotspot** command.

```
debug hotspot { events | packets } { enable | disable } { enable | disable }
```

Syntax Description	events	Configures debugging of HotSpot events.
	packets	Configures debugging of HotSpot packets.
	enable	Enables the debugging of HotSpot options.
	disable	Disables the debugging of HotSpot options.
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 hotspot events:

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

# debug ipv6

To configure debugging of IPv6 options, use the **debug ipv6** command.

**debug ipv6** {**all** | **bt** | **classifier** | **errors** | **events** | **filter** | **fsm** | **gleaner** | **hwapi** | **memory** | **ndsuppress** | **parser** | **policy** | **ra\_throttler** | **switcher**} {**enable** | **disable**}

## Syntax Description

<b>all</b>	Configures debugging of all the IPv6 information.
<b>bt</b>	Configures debugging of the IPv6 neighbor binding table.
<b>classifier</b>	Configures debugging of the IPv6 packet classifiers.
<b>errors</b>	Configures debugging of the IPv6 errors.
<b>events</b>	Configures debugging of the IPv6 events.
<b>filter</b>	Configures filters for the IPv6 debugs.
<b>fsm</b>	Configures debugging of the IPv6 finite state machine (FSM).
<b>gleaner</b>	Configures debugging of the IPv6 gleaner. Learning of entries is called gleaning.
<b>hwapi</b>	Configures debugging of the IPv6 hardware APIs.
<b>memory</b>	Configures debugging of the IPv6 binding table memory usage.
<b>ndsuppress</b>	Configures debugging of the suppressed IPv6 neighbor discoveries.
<b>parser</b>	Configures debugging of the IPv6 parser.
<b>policy</b>	Configures debugging of the IPv6 policies.
<b>ra_throttler</b>	Configures debugging of the IPv6 router advertising throttler.
<b>switcher</b>	Configures debugging of the IPv6 switcher.
<b>enable</b>	Enables debugging of the IPv6 options.
<b>disable</b>	Disables debugging of the IPv6 options.

## 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 IPv6 policies:

```
(Cisco Controller) >debug ipv6 policy enable
```