



## **Cisco Aironet Wave 2 and Catalyst Wi-Fi6 Access Point Command Reference, Release 8.10**

**First Published:** 2019-10-19

**Last Modified:** 2023-07-28

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019–2023 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### PREFACE

<b>Preface</b>	<b>ix</b>
Audience	<b>ix</b>
Document Conventions	<b>ix</b>
Related Documentation	<b>xii</b>
Communications, Services, and Additional Information	<b>xii</b>
Cisco Bug Search Tool	<b>xii</b>
Documentation Feedback	<b>xii</b>

---

### CHAPTER 1

<b>Using the Command Line Interface</b>	<b>1</b>
Understanding Command Modes	<b>1</b>
Understanding Abbreviated Commands	<b>2</b>
Understanding no Forms of Commands	<b>2</b>
Understanding CLI Error Messages	<b>2</b>
Configuring the Terminal	<b>3</b>
Recalling Commands	<b>4</b>
Accessing the CLI	<b>4</b>

---

### CHAPTER 2

<b>Supported Cisco Access Points</b>	<b>5</b>
--------------------------------------	----------

---

### CHAPTER 3

<b>capwap Commands</b>	<b>7</b>
capwap ap	<b>7</b>
capwap ap auth-token	<b>8</b>
capwap ap erase	<b>8</b>
capwap ap ethernet	<b>9</b>
capwap ap hostname	<b>9</b>
capwap ap ip	<b>10</b>

capwap ap lag 10  
capwap ap mesh strict-wired-uplink 11  
capwap ap mode 12  
capwap ap restart 12

---

**CHAPTER 4****clear Commands 13**

clear avc nbar 13  
clear counters 13  
clear cts 14  
clear datapath 15  
clear dot11 15  
clear logging 16

---

**CHAPTER 5****config Commands 17**

config ap address 17  
config ap client-trace 18  
config ap client-trace filter 19  
config ap client-trace output 20  
config boot baudrate 20  
config boot break 21  
config boot crashkernel 21  
config boot debug-memory 22  
config boot manual 22  
config boot path 23  
config cts debug enforcement host\_ip 23  
config cts debug enforcement rate 24  
config cts debug enforcement permissions 25  
config cts debug enforcement protocol 25

---

**CHAPTER 6****debug Commands 27**

debug arp 28  
debug ble 28  
debug capwap client 29  
debug capwap client avc 30

debug cdp	31
debug cleanair	31
debug dhcp	32
debug dot11 driver level	33
debug dot11 client data-path	33
debug dot11 client management	34
debug dot11 client probe	35
debug dot11 driver slot	35
debug dot11 firmware	36
debug dot11 sensor	37
debug dtls client	38
debug ethernet	38
debug flexconnect	39
debug lldp	40
debug memory	40
debug memory pool	41
debug memory pool alloc	41
debug memory pool free	42
debug mesh	43
debug mesh adjacency	43
debug mesh path-control	44
debug rrm neighbor	45
debug rrm reports	45
debug sip	46
debug wips	46
debug process memory	47
debug traffic	47
debug tunnel	48
debug client trace	48
no	49
traceroute	50
undebug	50

show ap client-trace status	54
show arp	55
show avc cft	55
show avc nbar	56
show avc netflow flows	56
show avc status	57
show boot	57
show capwap	58
show capwap client	59
show capwap client trace	59
show capwap ids sig	60
show cdp	60
show class-map	61
show cleanair debug	61
show client statistics	62
show clock	62
show configuration	63
show controller ble	63
show controllers dot11Radio	64
show controllers nss status	65
show controllers wired	66
show crypto	66
show debug	67
show dhcp	67
show dot11 qos	68
show dot11 wlan wpa3	68
show filesystems	69
show flash	69
show flexconnect	70
show flexconnect ocap firewall	70
show flexconnect wlan	71
show interfaces dot11Radio	72
show interfaces network	73
show interfaces wired	73

show inventory	74
show ip	74
show lacp	75
show logging	75
show memory	76
show policy-map	77
show processes	77
show processes memory	78
show rrm	79
show rrm rogue containment	80
show rrm rogue detection	81
show running-config	82
show security data-corruption	83
show security system state	83
show spectrum	84
show tech-support	85
show version	85
show trace dot11_chn	86
show trace	86
show wips	87

---

**CHAPTER 8****System Management Commands 89**

ap-type	89
archive	90
copy	90
delete	91
disable	92
enable	92
exec-timeout	92
logging	93
more	93
reload	94
terminal	95







## Preface

---

This preface describes the audience, organization, and conventions of the Cisco Aironet Wave 2 Access Point Command Reference. It also provides information about how to obtain other documentation.

- [Audience, on page ix](#)
- [Document Conventions, on page ix](#)
- [Related Documentation, on page xii](#)
- [Communications, Services, and Additional Information, on page xii](#)

## Audience

This publication is for experienced network administrators who configure and maintain Cisco Aironet Wave 2 Access Points.



---

**Note** Usage of **test** commands may cause system disruption such as unexpected reboot of the Cisco AP. Therefore, we recommend that you use the **test** commands on Cisco APs for debugging purposes with the help of Cisco Technical Assistance Center (TAC) personnel.

---

## Document Conventions

This document uses the following conventions:

Convention	Indication
<b>bold font</b>	Commands and keywords and user-entered text appear in <b>bold font</b> .
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	Elements in square brackets are optional.
{x   y   z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.

Convention	Indication
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
<code>courier font</code>	Terminal sessions and information the system displays appear in <code>courier font</code> .
<>	Nonprinting characters such as passwords are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.



**Note** Means reader take note. Notes contain helpful suggestions or references to material not covered in the manual.



**Tip** Means the following information will help you solve a problem.



**Caution** Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.



**Warning** This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. (To see translations of the warnings that appear in this publication, refer to the appendix "Translated Safety Warnings.")

Warning Title	Description
Waarschuwing	Dit waarschuwingssymbool betekent gevaar. U verkeert in een situatie die lichamelijk letsel kan veroorzaken. Voordat u aan enige apparatuur gaat werken, dient u zich bewust te zijn van de bij elektrische schakelingen betrokken risico's en dient u op de hoogte te zijn van standaard maatregelen om ongelukken te voorkomen. (Voor vertalingen van de waarschuwingen die in deze publicatie verschijnen, kunt u het aanhangsel "Translated Safety Warnings" (Vertalingen van veiligheidsvoorschriften) raadplegen.)
Varoitus	Tämä varoitusmerkki merkitsee vaaraa. Olet tilanteessa, joka voi johtaa ruumiinvammaan. Ennen kuin työskentelet minkään laitteiston parissa, ota selvää sähkökytkentöihin liittyvistä vaaroista ja tavanomaisista onnettomuuksien ehkäisykeinoista. (Tässä julkaisussa esiintyvien varoitusten käännökset löydät liitteestä "Translated Safety Warnings" (käännetyt turvallisuutta koskevat varoitukset).)

Warning Title	Description
Attention	Ce symbole d'avertissement indique un danger. Vous vous trouvez dans une situation pouvant entraîner des blessures. Avant d'accéder à cet équipement, soyez conscient des dangers posés par les circuits électriques et familiarisez-vous avec les procédures courantes de prévention des accidents. Pour obtenir les traductions des mises en garde figurant dans cette publication, veuillez consulter l'annexe intitulée « Translated Safety Warnings » (Traduction des avis de sécurité).
Warnung	Dieses Warnsymbol bedeutet Gefahr. Sie befinden sich in einer Situation, die zu einer Körperverletzung führen könnte. Bevor Sie mit der Arbeit an irgendeinem Gerät beginnen, seien Sie sich der mit elektrischen Stromkreisen verbundenen Gefahren und der Standardpraktiken zur Vermeidung von Unfällen bewusst. (Übersetzungen der in dieser Veröffentlichung enthaltenen Warnhinweise finden Sie im Anhang mit dem Titel "Translated Safety Warnings" (Übersetzung der Warnhinweise).)
Avvertenza	Questo simbolo di avvertenza indica un pericolo. Si è in una situazione che può causare infortuni. Prima di lavorare su qualsiasi apparecchiatura, occorre conoscere i pericoli relativi ai circuiti elettrici ed essere al corrente delle pratiche standard per la prevenzione di incidenti. La traduzione delle avvertenze riportate in questa pubblicazione si trova nell'appendice, "Translated Safety Warnings" (Traduzione delle avvertenze di sicurezza).
Advarsel	Dette varselsymbolet betyr fare. Du befinner deg i en situasjon som kan føre til personskade. Før du utfører arbeid på utstyr, må du være oppmerksom på de faremomentene som elektriske kretser innebærer, samt gjøre deg kjent med vanlig praksis når det gjelder å unngå ulykker. (Hvis du vil se oversettelser av de advarslene som finnes i denne publikasjonen, kan du se i vedlegget "Translated Safety Warnings" [Oversatte sikkerhetsadvarsler].)
Aviso	Este símbolo de aviso indica perigo. Encontra-se numa situação que lhe poderá causar danos físicos. Antes de começar a trabalhar com qualquer equipamento, familiarize-se com os perigos relacionados com circuitos eléctricos, e com quaisquer práticas comuns que possam prevenir possíveis acidentes. (Para ver as traduções dos avisos que constam desta publicação, consulte o apêndice "Translated Safety Warnings" - "Traduções dos Avisos de Segurança").
¡Advertencia!	Este símbolo de aviso significa peligro. Existe riesgo para su integridad física. Antes de manipular cualquier equipo, considerar los riesgos que entraña la corriente eléctrica y familiarizarse con los procedimientos estándar de prevención de accidentes. (Para ver traducciones de las advertencias que aparecen en esta publicación, consultar el apéndice titulado "Translated Safety Warnings.")
Varning	Denna varningssymbol signalerar fara. Du befinner dig i en situation som kan leda till personskada. Innan du utför arbete på någon utrustning måste du vara medveten om farorna med elkretsar och känna till vanligt förfarande för att förebygga skador. (Se förklaringar av de varningar som förekommer i denna publikation i appendix "Translated Safety Warnings" [Översatta säkerhetsvarningar].)

## Related Documentation

- Cisco Access Points—<https://www.cisco.com/c/en/us/products/wireless/access-points/index.html>
- Cisco Wireless Controller Software Documentation—<https://www.cisco.com/c/en/us/support/wireless/wireless-lan-controller-software/tsd-products-support-series-home.html>

## Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit [Cisco DevNet](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

## Cisco Bug Search Tool

[Cisco Bug Search Tool](#) (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

## Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.



## CHAPTER 1

# Using the Command Line Interface

This chapter describes the Cisco Aironet Wave 2 Access Point command-line interface (CLI) and how to use it to configure your AP.

- [Understanding Command Modes, on page 1](#)
- [Understanding Abbreviated Commands, on page 2](#)
- [Understanding no Forms of Commands, on page 2](#)
- [Understanding CLI Error Messages, on page 2](#)
- [Configuring the Terminal, on page 3](#)
- [Recalling Commands, on page 4](#)
- [Accessing the CLI, on page 4](#)

## Understanding Command Modes

The Cisco Aironet Wave 2 AP command line interface is divided into the following two different modes:

- **User EXEC mode**—When you start a session on the AP, you begin in the User EXEC mode. Only a limited subset of the commands are available in this mode. Also, the **show** commands that are available in the User EXEC mode are a subset of the **show** commands that are available in the Privileged EXEC mode.

The user EXEC commands are not saved when the AP is rebooted.

- **Privileged EXEC mode**—In this mode, you will have access to all commands. You are required to enter a password to enter the Privileged EXEC mode.

The commands available to you depend on which mode you are currently in. Enter a question mark (?) at the system prompt to obtain a list of commands available for the command mode you are in. For example, here are the list of User EXEC mode commands available:

```
cisco-wave2-ap>?  
Exec mode commands  
  enable  Turn on privileged commands  
  logout  Logout out from CLI  
  ping    Send echo messages  
  show    Show running system information
```

Table 1: Command Mode Summary

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Begin a session with your switch.	cisco-wave2-ap>	Enter <b>logout</b> or <b>quit</b> .	Use this mode to <ul style="list-style-type: none"> <li>• Change terminal settings.</li> <li>• Perform basic tests.</li> <li>• Display system information.</li> </ul>
Privileged EXEC	While in user EXEC mode, enter the <b>enable</b> command and enter the password when prompted.	cisco-wave2-ap#	Enter <b>disable</b> to exit.	Use this mode to verify commands that you have entered. Use a password to protect access to this mode.

## Understanding Abbreviated Commands

You need to enter only enough characters for the AP to recognize the command as unique.

This example shows how to enter the **show configuration** privileged EXEC command in an abbreviated form:

```
cisco-ap# show conf
```

## Understanding no Forms of Commands

While you need to use the **debug** command to enable debugs on many features, the prefix **no** disables debugs on those respective features. For example:

Command to enable debug:

```
cisco-ap# debug client ...
```

Command to disable debug:

```
cisco-ap# no debug client ...
```

## Understanding CLI Error Messages

This table lists some error messages that you might encounter while using the CLI to configure your AP.

Table 2: Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for your AP to recognize the command.	Enter the command again followed by a question mark (?) with a space between the command and the question mark.  The possible keywords that you can enter with the command appear.
% Incomplete command.	You did not enter all the keywords or values required by this command.	Enter the command again followed by a question mark (?) with a space between the command and the question mark.  The possible keywords that you can enter with the command appear.
% Invalid input detected at '^' marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode.  The possible keywords that you can enter with the command appear.

## Configuring the Terminal

### Before you begin

Enter the Privileged EXEC mode.

### Procedure

- Configure the number of lines on the screen by entering this command:

**terminal length** *number-of-lines*

Valid range is 0 to 512. If you enter 0, there will be no pausing.

#### Example:

```
cisco-ap# terminal length 20
```

- Copy debug output to the current terminal line by entering this command:

**terminal monitor**

- Disable logging to the current terminal line by entering this command:

**terminal monitor disable**

- Specify the terminal type by entering this command:

**terminal type** *type-name*

- Configure the number of characters that should be displayed on a screen line by entering this command:

**terminal width** *number-of-characters*

Valid range is 0 to 132.

**Example:**

```
cisco-ap# terminal width 30
```

## Recalling Commands

To recall commands from the history buffer, perform one of the actions listed in this table. These actions are optional.




---

**Note** The arrow keys function only on ANSI-compatible terminals such as VT100s.

---

*Table 3: Recalling Commands*

Action	Result
Press the up arrow key	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Press the down arrow key	Returns to more recent commands in the history buffer after recalling commands with the up arrow key. Repeat the key sequence to recall successively more recent commands.

## Accessing the CLI

You can access the CLI through a console connection, through Telnet, or by using the browser. Commands you enter in one session are not displayed in the other sessions. Therefore, it is possible to lose track of the session from which you entered commands.





## CHAPTER 2

# Supported Cisco Access Points

---

This book describes commands that are supported by the Cisco Aironet family of Access Points and Cisco Catalyst 9100 Wi-Fi6 family of Access Points.





## CHAPTER 3

# capwap Commands

- [capwap ap](#), on page 7
- [capwap ap auth-token](#), on page 8
- [capwap ap erase](#), on page 8
- [capwap ap ethernet](#), on page 9
- [capwap ap hostname](#), on page 9
- [capwap ap ip](#), on page 10
- [capwap ap lag](#), on page 10
- [capwap ap mesh strict-wired-uplink](#), on page 11
- [capwap ap mode](#), on page 12
- [capwap ap restart](#), on page 12

## capwap ap

To configure the primary, secondary and tertiary controllers for the AP, use the **capwap ap** command.

```
capwap ap {primary-base | secondary-base | tertiary-base}  
controller-name controller-ip-address
```

Syntax Description		
	<b>primary-base</b>	Configure AP's primary controller
	<b>secondary-base</b>	Configure AP's secondary controller
	<b>tertiary-base</b>	Configure AP's tertiary controller
	<i>controller-name</i>	Name of the controller
	<i>controller-ip-address</i>	IP address of the controller.

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to configure the primary controller for the AP:

```
cisco-ap# capwap ap primary-base wlc-5520 209.165.200.224
```

## capwap ap auth-token

To configure authentication token, use the **capwap ap auth-token** command.

```
capwap ap auth-token ssc-token
```

<b>Syntax Description</b>	<i>ssc-token</i> SSC token; valid range is 8 to 32 characters
<b>Command Modes</b>	Privileged EXEC (#)
<b>Command History</b>	<b>Release</b> <b>Modification</b>
	8.1.111.0 This command was introduced.

### Examples

The following example shows how to configure authentication token,:

```
cisco-ap# capwap ap auth-token myauthtoken
```

## capwap ap erase

To erase CAPWAP configuration, use the **capwap ap erase** command.

```
capwap ap erase {all | static-ip}
```

<b>Syntax Description</b>	<b>all</b> Erases all CAPWAP configuration
	<b>Note</b> If the AP is in Bridge mode, then the same Bridge mode is retained after the factory reset of the AP; if the AP is in FlexConnect, Local, Sniffer, or any other mode, then the AP mode is set to Local mode after the factory reset of the AP. If you press the Reset button on the AP and perform a true factory reset, then the AP moves to a cookie configured mode.
	<b>static-ip</b> Erase static IP or DNS configuration
<b>Command Modes</b>	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to erase all the CAPWAP configuration on the AP:

```
cisco-ap# capwap ap erase all
```

## capwap ap ethernet

To configure AP Ethernet parameters, use the **capwap ap ethernet** command.

**capwap ap ethernet tag** *ethernet-vlan-id*

Syntax Description	
	<i>ethernet-vlan-id</i> Ethernet VLAN ID; valid range is 0 to 4094. If you enter the VLAN ID value as 0, the VLAN tagging is disabled.

Command Modes	
	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to configure Ethernet VLAN tagging on the AP:

```
cisco-ap# capwap ap ethernet tag 2
```

## capwap ap hostname

To configure AP hostname, use the **capwap ap hostname** command.

**capwap ap hostname** *ap-name*

Syntax Description	
	<i>ap-name</i> AP name

Command Modes	
	Privileged EXEC (#)

Usage Guidelines	
	If the AP is already associated with a Cisco WLC, the new hostname is reflected on the Cisco WLC only after the AP dissociates and reassociates with the Cisco WLC.

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to configure a hostname for the AP:

```
cisco-ap# capwap ap hostname cisco-wave2-ap-2802
```

## capwap ap ip

To configure static IP address and DNS for the CAPWAP AP, use the **capwap ap ip** command.

```
capwap ap ip static-ip-addr static-netmask ip-addr-default-gateway [ip-addr-dns1 | ip-addr-dns2] [domain-name]
```

Syntax Description		
	<i>static-ip-addr</i>	Static IP address of the AP
	<i>static-netmask</i>	Static netmask
	<i>ip-addr-default-gateway</i>	IP address of the default gateway
	[ <i>ip-addr-dns1</i>   <i>ip-addr-dns2</i> ]	(Optional parameters) IP address(es) of the DNS
	[ <i>domain-name</i> ]	(Optional parameter) Domain name

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to configure static IP address and DNS for the CAPWAP AP:

```
cisco-ap# capwap ap ip 209.165.200.225 255.255.255.224 209.165.200.227 209.165.200.226 example.org
```

## capwap ap lag

To configure CAPWAP lag, use the **capwap ap lag** command.

```
capwap ap lag {enable | disable}
```

<b>Syntax Description</b>	<b>enable</b> Enables LAG
	<b>disable</b> Disables LAG

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

### Examples

The following example shows how to enable LAG on the AP:

```
cisco-ap# capwap ap lag enable
```

## capwap ap mesh strict-wired-uplink

To configure the root access points (RAPs) to stay as persistent RAPs even if the wired uplink is lost, use the **capwap ap mesh strict-wired-uplink** command.

**capwap ap mesh strict-wired-uplink** {enable | disable}

<b>Syntax Description</b>	<b>enable</b> Enables strict wired uplink on the Cisco AP.
	<b>disable</b> Disables strict wired uplink on the Cisco AP.

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	8.9 Cisco IOS XE Gibraltar 16.11.1	This command was introduced.

### Examples

The following example shows how to enable the root access points (RAPs) to stay as persistent RAPs even if the wired uplink is lost:

```
cisco-ap# capwap ap mesh strict-wired-uplink enable
```

## capwap ap mode

To configure AP mode, use the **capwap ap mode** command.

```
capwap ap mode { bridge | local }
```

<b>Syntax Description</b>	<b>bridge</b> Enables bridge mode
	<b>local</b> Enables local mode
<b>Command Modes</b>	Privileged EXEC (#)
<b>Command History</b>	<b>Release Modification</b>
	8.1.111.0 This command was introduced.

### Examples

The following example shows how to configure the AP to operate in local mode:

```
cisco-ap# capwap ap mode local
```

## capwap ap restart

To restart the CAPWAP protocol, use the **capwap ap restart** command.

```
capwap ap restart
```

<b>Syntax Description</b>	<b>restart</b> Restart the CAPWAP protocol
<b>Command Modes</b>	Privileged EXEC (#)
<b>Command History</b>	<b>Release Modification</b>
	8.1.111.0 This command was introduced.

### Examples

The following example shows how to restart CAPWAP protocol:

```
cisco-ap# capwap ap restart
```





## CHAPTER 4

# clear Commands

- [clear avc nbar](#), on page 13
- [clear counters](#), on page 13
- [clear cts](#), on page 14
- [clear datapath](#), on page 15
- [clear dot11](#) , on page 15
- [clear logging](#), on page 16

## clear avc nbar

To clear AVC NBAR statistics, use the **clear avc nbar** command.

### clear avc nbar statistics

<b>Syntax Description</b>	<b>statistics</b> Clears AVC NBAR statistics				
<b>Command Modes</b>	Privileged EXEC (#)				
<b>Command History</b>	<table><thead><tr><th>Release</th><th>Modification</th></tr></thead><tbody><tr><td>8.1.111.0</td><td>This command was introduced.</td></tr></tbody></table>	Release	Modification	8.1.111.0	This command was introduced.
Release	Modification				
8.1.111.0	This command was introduced.				

### Examples

The following example shows how to clear AVC NBAR statistics:

```
cisco-ap# clear avc nbar statistics
```

## clear counters

To clear 802.11 radio statistics, use the **clear counters** command.

**clear countersDot11Radio** *interface-number* | **client** | **fast-path profinet** | **wired** *interface-number*  
**MIB-stats**

### Syntax Description

<b>Dot11Radio</b>	(Optional) Clears the Dot11 interface statistics.
<i>interface-number</i>	Dot11Radio interface number; valid value is 0 or 1.
<b>client</b>	Clears the client statistics.
<b>fast-path</b>	Clears the controller fast-path statistics.
<b>profinet</b>	Clears the profinet statistics.
<b>wired</b>	Clears the wired interface statistics.
<i>interface-number</i>	Wired interface number, valid value is between 0 and 3.
<b>MIB-stats</b>	Clears the AP Internal-Switch MIB counters.

### Command Modes

Privileged EXEC (#)

### Command History

#### Release Modification

8.1.111.0	This command was introduced.
8.7	This command was enhanced by adding <b>client</b> , <b>fast-path</b> , <b>profinet</b> , <b>wired</b> parameters.

### Examples

The following example shows how to clear 802.11 interface statistics for the interface number specified:

```
cisco-ap# clear counters Dot11Radio 1
```

## clear cts

To clear the statistics of Cisco TrustSec Security, use the **clear cts** command.

**clear cts role-based counters** [**all** | **client** *mac-addr* | **from** *sgt* **to** *dgt*]

### Syntax Description

<b>counters</b>	Clears Cisco TrustSec summary counters
<b>all</b>	Clears all Cisco TrustSec counters
<b>client</b> <i>mac-addr</i>	Clears the Cisco TrustSec counters for a client MAC address specified in xx:xx:xx:xx:xx:xx format
<b>from</b>	Specifies the source group tag for filtered traffic
<i>sgt</i>	Security Group Tag (SGT); valid values are 0 to 65535

<b>to</b>	Specifies the destination group tag for filtered traffic
<i>dgt</i>	Destination Group Tag (DGT); valid values are 0 to 65535

**Command Modes** Privileged EXEC (#)

**Command History** **Release** **Modification**

8.1.111.0 This command was introduced.

This example shows you how to clear all the statistics of Cisco TrustSec Security counters:

```
cisco-ap# clear cts role-based counters all
```

## clear datapath

To clear the datapath counters or drops, use the **clear datapath** command.

**clear datapath** {drops | statistics}

**Syntax Description** **drops** Clears the datapath drop counters

**statistics** Clears the datapath counters

**Command Modes** Privileged EXEC (#)

**Command History** **Release** **Modification**

8.1.111.0 This command was introduced.

This example shows you how to clear the datapath drop counters:

```
cisco-ap# clear datapath drops
```

## clear dot11

To clear the 802.11 configuration, use the **clear dot11** command.

**clear dot11 sensor**

**Syntax Description** **sensor** Clears the sensor configuration and reboots

**Command Modes** Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

This example shows you how to clear the 802.11 configuration:

```
cisco-ap# clear dot11 sensor
```

## clear logging

To clear the logging details, use the **clear logging** command.

**clear logging** [**capwap** | **message** | **warning**]

Syntax Description	
	<b>capwap</b> (Optional) Clears CAPWAP logging details
	<b>message</b> (Optional) Clears message logging details
	<b>warning</b> (Optional) Clears warnings logging details

Command Modes	
	Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

This example shows you how to clear the CAPWAP logging details:

```
cisco-ap# clear logging capwap
```



## CHAPTER 5

# config Commands

- [config ap address](#) , on page 17
- [config ap client-trace](#), on page 18
- [config ap client-trace filter](#), on page 19
- [config ap client-trace output](#), on page 20
- [config boot baudrate](#), on page 20
- [config boot break](#), on page 21
- [config boot crashkernel](#), on page 21
- [config boot debug-memory](#), on page 22
- [config boot manual](#), on page 22
- [config boot path](#), on page 23
- [config cts debug enforcement host\\_ip](#), on page 23
- [config cts debug enforcement rate](#), on page 24
- [config cts debug enforcement permissions](#), on page 25
- [config cts debug enforcement protocol](#), on page 25

## config ap address

To configure the AP IPv4 or IPv6 address, use the **config ap address** command.

```
config ap address ipv4 { dhcp | static { static-ip-addr static-netmask default-gateway-ip-addr | ipv6
{ auto-config { enable | disable } | dhcp | disable | link-local ipv6-addr | static ipv6-addr ipv6-prefix
gateway-ipv6-addr
```

Syntax Description		
	<b>ipv4</b>	Configure IPv4 address
	<b>ipv6</b>	Configure IPv6 address
	<b>auto-config</b>	Auto configure IPv6 address
	<b>dhcp</b>	Configure IPv6 DHCP
	<b>auto-config</b>	
	<b>auto-config</b>	

**Command Default** None.

**Command History**

**Release Modification**

This command was introduced.

**Usage Guidelines**

**Examples**

**Related Commands**

Command	Description

## config ap client-trace

To configure client trace on the access point, use the **config ap client-trace** command.

```
config ap client-trace {address {add | clear-all | delete} | all-clients {enable | disable} | filter {all
{enable | disable} | arp {enable | disable} | assoc {enable | disable} | auth {enable | disable} | dhcp
{enable | disable} | eap {enable | disable} | icmp {enable | disable} | ndp {enable | disable} | probe
{enable | disable}} | inline-mon {enable | disable} | output console-log | start | stop}
```

**Syntax Description**

**addresses** Configure clients to trace. Specify the MAC address of the client

**add** Specifies a client to trace

**clear-all** Delete all client traces on this access point

**delete** Deletes client address to be traced. Takes a client MAC address

**all-clients** Trace all clients

**enable** Enables trace for all clients

**disable** Disables trace for all clients

**filter** Sets filters for cleint tracing

**all** Traces all filters

**arp** Traces ARP packets

Use the **enable** or **disable** keyword to enable or disable this filter.

**assoc** Traces ASSOC packets

**auth** Traces auth packets

**dhcp** Traces DHCP packets

**eap** Traces EAP packets

<b>icmp</b>	Traces ICMP packets
<b>ndp</b>	Traces NDP packets
<b>probe</b>	Trace probe packets.
<b>inline-mon</b>	Enables or disables inline monitoring
<b>output</b>	Enables or disables logging to the console or log file
<i>console-log</i>	Specifies console log keyword
<b>start</b>	Starts client tracing
<b>stop</b>	Stops client tracking

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to start client tracing on the AP:

```
cisco-ap# config ap client-trace start
```

## config ap client-trace filter

To set filters for client trace, use the **config ap client-trace filter** command.

```
config ap client-trace filter { all [ disable | enable ] | arp [ disable | enable ] |
assoc [ disable | enable ] | auth [ disable | enable ] | dhcp [ disable | enable ] |
eap [ disable | enable ] | icmp [ disable | enable ] | ndp [ disable | enable ] }
```

Syntax Description	
<b>all</b>	Trace all filters
<b>arp</b>	Trace ARP packets
<b>assoc</b>	Trace ASSOC packets
<b>auth</b>	Trace auth packets
<b>dhcp</b>	Trace DHCP packets
<b>eap</b>	Trace EAP packets
<b>icmp</b>	Trace ICMP packets

---

**ndp** Trace NDP Packets

---

**Command Modes** Privileged EXEC (#)

**Command History** **Release** **Modification**

---

8.1.111.0 This command was introduced.

---

To set filters for client trace, use this command:

```
cisco-ap# config ap client-trace filter
```

## config ap client-trace output

To configure the trace output, use the **config ap client-trace output** command.

```
config ap client-trace output console-log {disable | enable}
```

Syntax Description	console-log	Displays trace output to console and log
	disable	Disables trace output to console and log
	enable	Enables trace output to console and log

**Command Modes** Privileged EXEC (#)

**Command History** **Release** **Modification**

---

8.1.111.0 This command was introduced.

---

The following example shows you how to configure the trace output:

```
cisco-ap# config ap client-trace output
```

## config boot baudrate

To set the baud rate, use the **config boot baudrate** command.

```
config boot baudrate {115200 | 9600}
```

Syntax Description	115200	Sets the baud rate to 115200
	9600	Sets the baud rate to 9600

---



<b>Command Default</b>	The default config boot baud rate is 9600.
------------------------	--

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

### Examples

The following example shows how to configure the baud rate to 9600:

```
cisco-ap# config boot baudrate 9600
```

## config boot break

To enable break, use the **config boot break** command.

**config boot break** {enable | disable}

<b>Syntax Description</b>	<b>enable</b> Enables boot break
	<b>disable</b> Disables boot break

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

### Examples

The following example shows how to enable boot break:

```
cisco-ap# config boot break enable
```

## config boot crashkernel

To enable or disable kernel crash, use the **config boot crashkernel** command.

**config boot crashkernel** {enable | disable}

<b>Syntax Description</b>	<b>enable</b> Enables kernel crash
---------------------------	------------------------------------

---

**disable** Disables kernel crash

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

**Examples**

The following example shows how to enable kernel crash:

```
cisco-ap# config boot crashkernel enable
```

## config boot debug-memory

To enable memory debug, use the **config boot debug-memory** command.

**config boot debug-memory** {enable | disable}

**Syntax Description**


---

**enable** Enables memory debug

---

**disable** Disables memory debug

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

This example shows you how to enable memory debug:

```
cisco-ap# config boot debug-memory enable
```

## config boot manual

To enable manual boot of the AP, use the **config boot manual** command.

**config boot manual** {enable | disable}

**Syntax Description**


---

**enable** Enables manual boot

---

---

**disable** Disables manual boot

---



---

**Command Modes** Privileged EXEC (#)

---



---

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to enable manual boot:

```
cisco-ap# config boot manual enable
```

## config boot path

To configure the boot path, use the **config boot path** command.

```
config boot path {1 | 2}
```

---

**Syntax Description**

{1   2}	Path to be specified as Part 1 or Part 2
---------	--

---



---

**Command Modes** Privileged EXEC (#)

---



---

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to configure the booth path as 1:

```
cisco-ap# config boot path 1
```

## config cts debug enforcement host\_ip

To filter the SGACL enforcement debugs based on the host IP, use the **config cts debug enforcement host\_ip** command.

```
config cts debug enforcement host_ip {ipv4 dst-ip [src-ip] | ipv6 dst-ip [src-ip]}
```

<b>Syntax Description</b>	<b>ipv4</b> <i>dst-ip</i> [ <i>src-ip</i> ] Displays only the IPv4 SGACL enforcement debugs based on the destination and, optionally, source IP addresses
	<b>ipv6</b> <i>dst-ip</i> [ <i>src-ip</i> ] Displays only the IPv6 SGACL enforcement debugs based on the destination and, optionally, source IP addresses

**Command Modes** Privileged EXEC (#)

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

The following example shows you how to filter the IPv4 SGACL enforcement debugs based on the host IP:

```
cisco-ap# config cts debug enforcement host_ip ipv4 209.165.200.224 209.165.200.227
```

## config cts debug enforcement rate

To configure the rate of printing of debug logs, use the **config cts debug enforcement rate** command.

**config cts debug enforcement rate** {*X Y*}

**Command Modes** Privileged EXEC (#)

**Syntax Description**

**rate** Configure the rate of printing debug logs

*X* Number of packets whose debugs are to be displayed for every *Y* number of packets processed; valid range is between 0 to 10000

*Y* Number of packets to be processed; valid range is between 0 to 10000

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

### Examples

The following example shows how to configure the rate of printing of debug logs such that debugs of 100 packets are displayed for every 500 packets processed:

```
cisco-ap# config cts debug enforcement rate 100 500
```

## config cts debug enforcement permissions

To filter SGACL enforcement debugs based on source group tag (SGT) and destination group tag (DGT), use the **config cts debug enforcement permissions** command.

```
config cts debug enforcement permissions { dgt | sgt } tag-id
```

<b>Syntax Description</b>	<b>dgt</b> Destination group tag
	<b>sgt</b> Source group tag
	<i>tag-id</i> Tag identifier; valid values are between 0 to 65535

**Command Modes** Privileged EXEC (#)

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

The following example shows you how to filter SGACL enforcement debugs for a destination group tag whose ID is 600:

```
cisco-ap# config cts debug enforcement permissions dgt 600
```

## config cts debug enforcement protocol

To filter SGACL enforcement debugs based on protocol, use the **config cts debug enforcement protocol** command.

```
config cts debug enforcement protocol {protocol-id | icmp | tcp | udp}
```

<b>Syntax Description</b>	<i>protocol-id</i> Protocol ID; valid values are between 0 to 65535
	<b>icmp</b> Filter SGACL enforcement for ICMP traffic
	<b>tcp</b> Filter SGACL enforcement for TCP traffic
	<b>udp</b> Filter SGACL enforcement for UDP traffic

**Command Modes** Privileged EXEC (#)

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

The following example shows you how to filter SGACL enforcement debugs based on protocol for UDP traffic:

```
cisco-ap# config cts debug enforcement protocol udp
```



## CHAPTER 6

# debug Commands

---

- [debug arp](#), on page 28
- [debug ble](#), on page 28
- [debug capwap client](#), on page 29
- [debug capwap client avc](#), on page 30
- [debug cdp](#), on page 31
- [debug cleanair](#), on page 31
- [debug dhcp](#), on page 32
- [debug dot11 driver level](#), on page 33
- [debug dot11 client data-path](#), on page 33
- [debug dot11 client management](#), on page 34
- [debug dot11 client probe](#), on page 35
- [debug dot11 driver slot](#), on page 35
- [debug dot11 firmware](#), on page 36
- [debug dot11 sensor](#), on page 37
- [debug dtls client](#), on page 38
- [debug ethernet](#), on page 38
- [debug flexconnect](#), on page 39
- [debug lldp](#), on page 40
- [debug memory](#), on page 40
- [debug memory pool](#), on page 41
- [debug memory pool alloc](#), on page 41
- [debug memory pool free](#), on page 42
- [debug mesh](#), on page 43
- [debug mesh adjacency](#), on page 43
- [debug mesh path-control](#), on page 44
- [debug rrm neighbor](#), on page 45
- [debug rrm reports](#), on page 45
- [debug sip](#), on page 46
- [debug wips](#), on page 46
- [debug process memory](#), on page 47
- [debug traffic](#), on page 47
- [debug tunnel](#), on page 48
- [debug client trace](#), on page 48

- [no](#), on page 49
- [traceroute](#), on page 50
- [undebug](#), on page 50

## debug arp

To enable debugging of ARP, use the **debug arp** command.

**debug arp** {**errors** | **events** | **packets**}

Syntax Description	
<b>errors</b>	Enable debugging of ARP errors
<b>events</b>	Enable debugging of ARP events
<b>packets</b>	Enable debugging of ARP Tx and Rx packets

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

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

```
cisco-ap# debug arp errors
```

## debug ble

To enable debugging of Bluetooth Low Energy (BLE), use the **debug ble** command.

**debug ble** {**critical** | **error** | **events** | **fastpath** {**rss**i | **scan** | **sync**} | **receive** | **transmit**}

Syntax Description	
<b>critical</b>	Enables debugging of BLE critical events
<b>error</b>	Enables debugging of BLE error events
<b>events</b>	Enables debugging of BLE events
<b>fastpath</b> { <b>rss</b> i   <b>scan</b>   <b>sync</b> }	Shows data exported to CMX. The following options are available: <ul style="list-style-type: none"> <li>• RSSI data</li> <li>• Scan data</li> <li>• Sync data</li> </ul>



<b>receive</b>	Enables debugging of BLE packet received from BLE radio
<b>transmit</b>	Enables debugging of BLE packet transmitted to BLE radio

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.7	This command was introduced.
-----	------------------------------

**Examples**

The following example shows how to enable debugging of BLE critical events:

```
cisco-ap# debug ble critical
```

## debug capwap client

To enable debugging of CAPWAP clients, use the **debug capwap client** command.

**debug capwap client** { **ble** | **detail** | **efficient-upgrade** | **error** | **events** | **flexconnect** | **info** | **keepalive** | **payload** | **pmtu** | **qos** | **reassemble** | **security** }

**Syntax Description**

<b>ble</b>	Enables debugging of CAPWAP BLE detail
<b>detail</b>	Enables debugging of CAPWAP detail
<b>efficient-upgrade</b>	Enables debugging of image predownload
<b>error</b>	Enables debugging of CAPWAP error
<b>events</b>	Enables debugging of CAPWAP events
<b>flexconnect</b>	Enables debugging of CAPWAP FlexConnect mode event
<b>info</b>	Enables debugging of CAPWAP information
<b>keepalive</b>	Enables debugging of CAPWAP keepalive
<b>payload</b>	Enables debugging of CAPWAP payload
<b>pmtu</b>	Enables debugging of CAPWAP path MTU
<b>qos</b>	Enables debugging of CAPWAP QoS
<b>reassemble</b>	Enables debugging of CAPWAP reassembly
<b>security</b>	Enables debugging of CAPWAP security

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

### Examples

The following example shows how to enable debugging of CAPWAP client detail:

```
cisco-ap# debug capwap client detail
```

## debug capwap client avc

To enable debugging of CAPWAP client AVC, use the **debug capwap client avc** command.

**debug capwap client avc** {all | detail | error | event | info | netflow} {all | detail | error | event | packet} | numflows}

<b>Syntax Description</b>		
<b>all</b>	Enables debugging of all CAPWAP client AVC	
<b>detail</b>	Enables debugging of CAPWAP AVC detail	
<b>error</b>	Enables debugging of CAPWAP AVC error	
<b>event</b>	Enables debugging of CAPWAP AVC event	
<b>info</b>	Enables debugging of CAPWAP AVC information	
<b>netflow</b>	Enables debugging of CAPWAP client AVC NetFlow	
<b>netflow all</b>	Enables debugging of all CAPWAP client AVC NetFlow	
<b>netflow detail</b>	Enables debugging of CAPWAP client AVC NetFlow detail	
<b>netflow error</b>	Enables debugging of CAPWAP client AVC NetFlow error	
<b>netflow event</b>	Enables debugging of CAPWAP client AVC NetFlow event	
<b>netflow packet</b>	Enables debugging of CAPWAP client AVC NetFlow packet	
<b>numflows</b>	Enables debugging of CAPWAP client AVC numflows	

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

**Examples**

The following example shows how to enable debugging of all CAPWAP client AVC:

```
cisco-ap# debug capwap client avc all
```

## debug cdp

To enable debugging of controller discovery protocol (CDP), use the **debug cdp** command.

```
debug cdp {adjacency | events | ilp | packets}
```

**Syntax Description**

<b>adjacency</b>	Enables debugging of CDP neighbors
<b>events</b>	Enables debugging of CDP events
<b>ilp</b>	Enables debugging of inline power
<b>packets</b>	Enables debugging of CDP packets

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

**Examples**

The following example shows how to enable debugging of CDP events:

```
cisco-ap# debug cdp events
```

## debug cleanair

To configure debugging of CleanAir, use the **debug cleanair** command.

```
debug cleanair {bringup | event | logdebuglow | major | nsi | offchan {0 | 1}}
```

**Syntax Description**

<b>bringup</b>	Enables debugging of CleanAir port or bringups
<b>events</b>	Enables debugging of normal CleanAir events
<b>logdebug</b>	Logs CleanAir debug output to a logfile
<b>low</b>	Enables debugging of hex dump of some messages

<b>major</b>	Enables debugging of major CleanAir events
<b>nsi</b>	Enables debugging of NSI messages
<b>offchan 0   1</b>	Enables debugging of CleanAir MSMT requests. You have to specify the radio slot as either 0 or 1

**Command Modes** Privileged EXEC (#)

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

**Examples**

The following example shows how to enable debugging of major CleanAir events:

```
cisco-ap# debug cleanair major
```

## debug dhcp

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

**debug dhcp {errors | events | packets}**

**Syntax Description**

**errors** Enables debugging of DHCP errors

**events** Enables debugging of DHCP events

**packets** Enables debugging of DHCP packets

**Command Modes** Privileged EXEC (#)

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

**Examples**

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

```
cisco-ap# debug dhcp errors
```

## debug dot11 driver level

To enable debugging of 802.11, use the **debug dot11 driver level** command.

```
debug dot11 driver level { critical | errors | events | info }
```

Syntax Description	
<b>critical</b>	Enables 802.11 critical level debugging
<b>errors</b>	Enables 802.11 error level debugging
<b>events</b>	Enables 802.11 event level debugging
<b>info</b>	Enables 802.11 information level debugging

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of 802.11 error level:

```
cisco-ap# debug dot11 driver level errors
```

## debug dot11 client data-path

To enable debugging of 802.11 client data-path, use the **debug dot11 client data-path** command.

```
debug dot11 client data-path {{ all-types | arp | dhcp | eapol | ipv6-ra |.opendns | dns-acl }} { addr { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 }}
```

Syntax Description	
<b>arp</b>	Enables client datapath ARP debugging
<b>dhcp</b>	Enables client datapath DHCP debugging
<b>eapol</b>	Enables client datapath EAPOL debugging
<b>dns-acl</b>	Enables client datapath DNS-ACL debugging
<b>ipv6-ra</b>	Enables client data-path IPv6 RA-MC2UC debugging
<b>opendns</b>	Enables client data-path openDNS debugging
<b>{addr   all-types}</b>	Option to specify MAC address of specific clients or all clients

---

{*mac-addr1* | *mac-addr2* | *mac-addr3* | *mac-addr4*} MAC addresses of clients that you have to enter  
| *mac-addr4*}

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

**Examples**

The following example shows how to enable debugging of client data-path ARP:

```
cisco-ap# debug dot11 client data-path arp
```

## debug dot11 client management

To enable 802.11 client debugging level, use the **debug dot11 client management** command.

```
debug dot11 client management { critical | errors | events | info } { addr { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 } }
```

**Syntax Description**

<b>critical</b>	Enables client critical level debugging
<b>errors</b>	Enables client error level debugging
<b>events</b>	Enables client event level debugging
<b>info</b>	Enables client information level debugging
{ <i>mac-addr1</i>   <i>mac-addr2</i>   <i>mac-addr3</i>   <i>mac-addr4</i> }	MAC addresses of clients that you have to enter

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

**Examples**

The following example shows how to enable debugging of a client at the event level:

```
cisco-ap# debug dot11 client management events e1:90:6f:7e:e6:29
```

## debug dot11 client probe

To enable 802.11 client debugging probe, use the **debug dot11 client probe** command.

```
debug dot11 client probe { { address mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 } | all }
```

### Syntax Description

<b>address</b>	Probe specific clients using their MAC addresses.
<i>mac-addr</i>	MAC addresses of the clients. You can enter upto four MAC addresses.
<b>all</b>	Probe all the clients associated with the AP.

### Command Modes

Privileged EXEC (#)

### Command History

Release	Modification
8.10	This command was introduced.

### Example

The following example shows how to enable debugging of all clients:

```
cisco-wave2-ap# debug dot11 client probe all
```

## debug dot11 driver slot

To enable debugging of 802.11 drivers, use the **debug dot11 driver slot** command.

```
debug dot11 driver slot { 0 | 1 } { all-types | { cac { info | metrics } } | chd | save-accounting-data | save-on-failure [ extended ] | stop-on-failure | metrics traffic | metrics video | type { all | association | authentication | dhcp | eap | icmp | probe } mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4
```

### Syntax Description

<b>slot</b> { <i>0</i>   <i>1</i> }	Enables 802.11 driver debugs per radio
<b>all-types</b>	Enables all 802.11 driver debugs
<b>cac</b>	Enables 802.11 CAC debugs
<b>cac info</b>	Enables 802.11 CAC info level debugs
<b>cac metrics</b>	Enables debugging of 802.11 CAC metrics
<b>chd</b>	Enables 802.11 CHD debugs
<b>save-accounting-data</b>	Saves the radio accounting data

<b>save-on-failure</b>	Saves the radio crash information upon radio failure
<b>save-on-failure extended</b>	Saves extended information on radio failure
<b>stop-on-failure</b>	Stops the AP from reboot on radio failure
<b>metrics traffic</b>	Enables 802.11 traffic stream metric debugs
<b>metrics video</b>	Enables 802.11 video metric debugs
<b>type</b>	Enables the debug types.
<b>all</b>	Enables the all type debugging.
<b>association</b>	Enables the association debugging.
<b>authentication</b>	Enables the authentication debugging.
<b>dhcp</b>	Enables the dhcp debugging.
<b>eap</b>	Enables the eap debugging.
<b>icmp</b>	Enables the icmp debugging.
<b>probe</b>	Enables the probe debugging.
<i>mac-addr</i>	MAC addresses of the clients. You can enter upto four MAC addresses.

**Command Modes**

Privileged EXEC (#)

**Command History**

Release	Modification
8.1.111.0	This command was introduced.
8.5.140.0 and 8.8	This command was enhanced by adding the <b>type</b> parameter.

**Examples**

The following example shows how to enable debugging of CAC at the information level:

```
cisco-ap# debug dot11 driver slot cac info
```

## debug dot11 firmware

To debug the 802.11 firmware, use the **debug dot11 firmware** command.

```
debug dot11 firmware slot slot_ID level { all-level | critical | emergency | error | info }
address { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 }
```

**Syntax Description**

<i>slot_ID</i>	Enables 802.11 driver debugs per radio
----------------	--



<b>all-level</b>	Enables all the debug levels.
<b>critical</b>	Enables critical level debugs.
<b>emergency</b>	Enables emergency level debugs.
<b>error</b>	Enables error level debugs.
<b>info</b>	Enables info level debugs.
<b>address</b>	To add client address for driver/firmware debugging.
<b>mac-addr</b>	MAC addresses of the clients. You can enter upto four MAC addresses.

**Command Modes**

Privileged EXEC (#)

**Command History**

Release	Modification
8.5.140.0 and 8.8	This command was introduced.

**Example**

The following example shows how to enable debugging of 802.11 emergency level:

```
cisco-wave2-ap# debug dot11 firmware slot 1 emergency address 92:FB:D6:B3:7A:6C
```

## debug dot11 sensor

To enable debugging of 802.11 sensors, use the **debug dot11 sensor** command.

```
debug dot11 sensor {dns | file-transfer | mail-server | ping | radius | ssh | telnet | web-server }
```

**Syntax Description**

<b>dns</b>	Enables debugging of 802.11 sensor DNS
<b>file-transfer</b>	Enables debugging of 802.11 sensor file transfer
<b>mail-server</b>	Enables debugging of 802.11 sensor mail server
<b>ping</b>	Enables debugging of 802.11 sensor ping
<b>radius</b>	Enables debugging of 802.11 sensor radius
<b>ssh</b>	Enables debugging of 802.11 sensor SSH
<b>telnet</b>	Enables debugging of 802.11 sensor Telnet.
<b>web-server</b>	Enables debugging of 802.11 sensor web server

---

**Command Modes** Privileged EXEC (#)

---

Command History	Release	Modification
	8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to enable debugging of 802.11 sensor file transfer:

```
cisco-ap# debug dot11 sensor file-transfer
```

## debug dtls client

To configure DTLS client error and event debugging, use the **debug dtls client** command.

```
debug dtls client { error | event [detail] }
```

Syntax Description	error	event [detail]
	Configures debugging of DTLS client errors	Configures debugging of DTLS client events

---



---

**Command Modes** Privileged EXEC (#)

---

Command History	Release	Modification
	8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to enable debugging of DTLS client events:

```
cisco-ap# debug dtls client event
```

## debug ethernet

To configure Ethernet debugging, use the **debug ethernet** command.

```
debug ethernet interface-number { both | rcv | xmt }
```

Syntax Description	<i>interface-number</i>	both
	Interface number that you have to enter as either 0 or 1	Enables debugging of both transmission and reception

---

<b>rcv</b>	Enables debugging of reception
<b>xmt</b>	Enables debugging of transmission

**Command Modes** Privileged EXEC (#)

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of transmission for interface 0:

```
cisco-ap# debug ethernet 0 xmt
```

## debug flexconnect

To debug FlexConnect features, use the **debug flexconnect** command.

**debug flexconnect** {acl | cckm | dot11r | event | multicast {igmp | traffic} | pmk | proxy-arp | vsa | wlan-vlan | wsastats}

Syntax Description	
<b>acl</b>	Configures debugging of FlexConnect ACL
<b>cckm</b>	Configures debugging of CCKM
<b>dot11r</b>	Configures debugging of 802.11r
<b>event</b>	Configures debugging of wireless control protocol (WCP) events
<b>multicast igmp</b>	Configures debugging of Multicast IGMP
<b>multicast traffic</b>	Configures debugging of Multicast traffic
<b>pmk</b>	Configures debugging of opportunistic key caching (OKC) or pairwise master key caching
<b>vsa</b>	Configures debugging of AAA vendor specific attributes (VSA)
<b>wlan-vlan</b>	Configures debugging of WLAN-VLAN mapping
<b>wsastats</b>	Configures debugging of RADIUS or DHCP wireless service assurance statistics

**Command Modes** Privileged EXEC (#)

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

**Examples**

The following example shows how to enable debugging of FlexConnect ACL:

```
cisco-ap# debug flexconnect acl
```

## debug lldp

To debug LLDP, use the **debug lldp** command.

```
debug lldp {errors | events | packet}
```

Syntax Description	
<b>errors</b>	Debugs LLDP errors
<b>events</b>	Debugs LLDP events
<b>packet</b>	Debugs LLDP packets

Command Modes	Privileged EXEC (#)
---------------	---------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

**Examples**

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

```
cisco-ap# debug lldp errors
```

## debug memory

To debug memory, use the **debug memory** command.

```
debug memory {clear | save}
```

Syntax Description	
<b>clear</b>	Removes memory debug upon boot-up
<b>save</b>	Saves current debug level and applies it upon following boots

Command Modes	Privileged EXEC (#)
---------------	---------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to remove memory debug upon boot-up:

```
cisco-ap# debug memory clear
```

## debug memory pool

To debug memory pool, use the **debug memory pool** command.

```
debug memory pool {diff | realtime interval 1-1000000-seconds | start}
```

Syntax Description	diff	Shows memory pool debug difference in detail
	<b>realtime interval</b> <i>1-1000000-seconds</i>	Configures realtime interval for the memory pool
	<b>start</b>	Starts the debug for the memory pool

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to configure realtime interval of 180 seconds for the memory pool:

```
cisco-ap# debug memory pool realtime interval 180
```

## debug memory pool alloc

To debug memory pool allocation calls, use the **debug memory pool alloc** command.

```
debug memory pool alloc {all | name pool-name} {diff | realtime interval 1-1000000-seconds | start}
```

Syntax Description	all	Configures debug for all memory pool allocation calls
	<b>name</b> <i>pool-name</i>	Configures debug for a specific memory pool's allocation call

<b>diff</b>	Shows memory pool debug allocation call difference in detail
<b>realtime interval</b> <i>1-1000000-seconds</i>	Configures realtime interval for the memory pool allocation calls
<b>start</b>	Starts the debug for the memory pool allocation calls

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

**Examples**

The following example shows how to configure the start of the debug for all memory pool allocation calls:

```
cisco-ap# debug memory pool alloc all start
```

## debug memory pool free

To debug memory pool free calls, use the **debug memory pool free** command.

```
debug memory pool free { all | name pool-name } { diff | realtime interval 1-1000000-seconds | start }
```

**Syntax Description**

<b>all</b>	Configures debug for all memory pool free calls
<b>name</b> <i>pool-name</i>	Configures debug for a specific memory pool's free call
<b>diff</b>	Shows memory pool debug free call difference in detail
<b>realtime interval</b> <i>1-1000000-seconds</i>	Configures realtime interval for the memory pool free calls
<b>start</b>	Starts the debug for the memory pool free calls

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

**Examples**

The following example shows how to configure the start of the debugging of all memory pool free calls:

```
cisco-ap# debug memory pool free all start
```

## debug mesh

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

```
debug mesh {channel | clear | convergence | events | forward-mcast | forward-packet | forward-table | linktest | path-control | port-control | security | trace}
```

### Syntax Description

<b>channel</b>	Configures debugging of mesh channel
<b>clear</b>	Resets all mesh debugs
<b>convergence</b>	Configures debugging of mesh convergence
<b>events</b>	Configures debugging of mesh events
<b>forward-mcast</b>	Configures debugging of mesh forwarding Multicast
<b>forward-packet</b>	Configures debugging of mesh forwarding packets
<b>forward-table</b>	Configures debugging of mesh forwarding table
<b>linktest</b>	Configures debugging of mesh linktest
<b>port-control</b>	Configures debugging of mesh port control
<b>security</b>	Configures debugging of mesh security
<b>trace</b>	Configures debugging of mesh trace

### Command Modes

Privileged EXEC (#)

### Command History

Release	Modification
8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of mesh channel:

```
cisco-ap# debug mesh channel
```

## debug mesh adjacency

To debug mesh adjacency, use the **debug mesh adjacency** command.

```
debug mesh adjacency {child | clear | dfs | message | packet | parent }
```

Syntax Description	
<b>adjacency</b>	Debug mesh adjacency
<b>child</b>	Debug mesh adjacency child
<b>clear</b>	Debug clear mesh adjacency
<b>dfs</b>	Debug mesh DFS
<b>message</b>	Debug mesh adjacency messages
<b>packet</b>	Debug mesh adjacency packet
<b>parent</b>	Debug mesh adjacency parent

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of mesh adjacency parent:

```
cisco-ap# debug mesh adjacency parent
```

## debug mesh path-control

To configure debugging of mesh path control, use the **debug mesh path-control** command.

```
debug mesh path-control {error | events | packets}
```

Syntax Description	
<b>error</b>	Configures debugging of mesh path control errors
<b>events</b>	Configures debugging of mesh path control events
<b>packets</b>	Configures debugging of mesh path control packets

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of mesh path control errors:



```
cisco-ap# debug mesh path-control error
```

## debug rrm neighbor

To enable RRM neighbor debugging, use the **debug rrm neighbor** command.

```
debug rrm neighbor {tx | rx | detail }
```

Syntax Description	tx	Enable RRM neighbor Tx debugging
	rx	Enable RRM neighbor Rx debugging
	detail	Enable RRM neighbor detail debugging

Command Modes	Privileged EXEC (#)
---------------	---------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of RRM neighbor transmissions:

```
cisco-ap# debug rrm neighbor tx
```

## debug rrm reports

To enable RRM reports debugging, use the **debug rrm reports** command.

```
debug rrm reports
```

Syntax Description	reports	Enables RRM report debugging
--------------------	---------	------------------------------

Command Modes	Privileged EXEC (#)
---------------	---------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable debugging of RRM reports:

```
cisco-ap# debug rrm reports
```

## debug sip

To enable session initiation protocol (SIP) debugging, use the **debug sip** command.

```
debug sip {all | tx | rx}
```

Syntax Description	
	<b>all</b> Enabling SIP transmission and reception debugging
	<b>tx</b> Enabling SIP transmission debugging
	<b>rx</b> Enabling SIP reception debugging

Command Modes	
	Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

### Examples

The following example shows how to enable debugging of SIP transmissions and reception:

```
cisco-ap# debug sip all
```

## debug wips

To enable wIPS debugging, use the **debug wips** command.

```
debug wips {errors | events | critical}
```

Syntax Description	
	<b>errors</b> Enable wIPS error level debugging
	<b>events</b> Enable wIPS event level debugging
	<b>critical</b> Enable wIPS critical level debugging

Command Modes	
	Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

### Examples

The following example shows how to enable wIPS error level debugging:

```
cisco-ap# debug wips errors
```

## debug process memory

To process memory debugging, use the **debug process memory** command.

```
debug process memory {diff | realtime [interval interval-in-seconds ] | start}
```

### Syntax Description

<b>diff</b>	Process memory debug show diff
<b>realtime</b>	Process memory real time debug
<b>interval</b>	Update interval; valid range 1 to 1000000 seconds
<b>start</b>	Process memory debug start

### Command Modes

Privileged EXEC (#)

### Command History

#### Release Modification

8.1.111.0	This command was introduced.
-----------	------------------------------

### Examples

The following example shows how to enable the start of debugging of process memory:

```
cisco-ap# debug process memory start
```

## debug traffic

To enable traffic debugging, use the **debug traffic** command.

```
debug traffic {host {icmpv6 | ip | ipv6 | tcp | udp { verbose}}} | wired {ip | tcp | udp { verbose}}}
```

### Syntax Description

<b>host</b>	Enabling host traffic debugging
<b>wired</b>	Enabling wired traffic debugging
<b>verbose</b>	Display verbose output
<b>icmpv6</b>	Enabling host ICMPv6 traffic dump

---

<b>ip</b>	Enabling host IP traffic dump
-----------	-------------------------------

---

<b>ipv6</b>	Enabling host IPv6 traffic dump
-------------	---------------------------------

---

<b>tcp</b>	Enabling TCP traffic dump
------------	---------------------------

---

<b>udp</b>	Enabling UDP traffic dump
------------	---------------------------

---



---

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

---

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

---

### Examples

The following example shows how to enable debugging of host IP traffic dump:

```
cisco-ap# debug traffic host ip
```

## debug tunnel

To configure debugging of tunnel, use the **debug tunnel** command.

### debug tunnel eogre

---

<b>Syntax Description</b>	<b>eogre</b> Configures debugging of EoGRE tunnel
---------------------------	---

---



---

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

---

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

---

### Examples

The following example shows how to enable debugging of EoGRE tunnel:

```
cisco-ap# debug tunnel eogre
```

## debug client trace

To enable client trace debugging, use the **debug client trace** command.

**debug client trace** {**all** | **address** *mac-address* | **enable** | **filter** {**assoc** | **auth** | **dhcp** | **eap** | **icmp** | **mgmt** | **probe** | **proto**}}

Syntax Description	all	Configure all clients tracing
	<b>address</b>	Configure address(es) to trace
	<i>mac-address</i>	MAC address to trace
	<b>enable</b>	Enable tracing
	<b>filter</b>	Configure trace filter
	<b>assoc</b>	Trace Association packets
	<b>auth</b>	Trace Authentication packets
	<b>dhcp</b>	Trace DHCP packets
	<b>eap</b>	Trace EAP packets
	<b>icmp</b>	Trace ICMP packets
	<b>mgmt</b>	Trace probe, assoc, auth, EAP packets
	<b>probe</b>	Trace probe packets
	<b>proto</b>	Trace DHCP, ICMP packets

**Command Modes** Privileged EXEC (#)

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

### Examples

The following example shows how to enable tracing of all clients:

```
cisco-ap# debug client trace all
```

## no

To negate a command or set to its defaults, use the **no** command.

**no**

**Command Modes** Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

To negate a command or set to its defaults, use this command:

```
cisco-ap# no debug
```

## tracert

To view the routes followed by packets traveling in the network, use the **tracert** command.

**tracert** *destination-address*

**Syntax Description**

*destination-address* IP address of the destination of the packets

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

**Examples**

The following example shows how to view the routes followed by packets traveling in the network, with a destination IP address specified:

```
cisco-ap# tracert 209.165.200.224
```

## undeb

To disable debugging on the access point, use the **undeb** command.

**undeb** [**all**]

**Syntax Description**

**a** Disables all debugging messages.

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

### Examples

The following example shows how to disable all debugging messages:

```
cisco-ap# undebug all
```







## CHAPTER 7

# show Commands

---

- [show ap client-trace status](#), on page 54
- [show arp](#), on page 55
- [show avc cft](#), on page 55
- [show avc nbar](#), on page 56
- [show avc netflow flows](#), on page 56
- [show avc status](#), on page 57
- [show boot](#), on page 57
- [show capwap](#), on page 58
- [show capwap client](#), on page 59
- [show capwap client trace](#), on page 59
- [show capwap ids sig](#), on page 60
- [show cdp](#), on page 60
- [show class-map](#), on page 61
- [show cleanair debug](#), on page 61
- [show client statistics](#), on page 62
- [show clock](#), on page 62
- [show configuration](#), on page 63
- [show controller ble](#), on page 63
- [show controllers dot11Radio](#), on page 64
- [show controllers nss status](#), on page 65
- [show controllers wired](#), on page 66
- [show crypto](#), on page 66
- [show debug](#), on page 67
- [show dhcp](#), on page 67
- [show dot11 qos](#), on page 68
- [show dot11 wlan wpa3](#), on page 68
- [show filesystems](#), on page 69
- [show flash](#), on page 69
- [show flexconnect](#), on page 70
- [show flexconnect ocap firewall](#), on page 70
- [show flexconnect wlan](#), on page 71
- [show interfaces dot11Radio](#), on page 72
- [show interfaces network](#), on page 73

- [show interfaces wired](#), on page 73
- [show inventory](#), on page 74
- [show ip](#), on page 74
- [show lacp](#), on page 75
- [show logging](#), on page 75
- [show memory](#), on page 76
- [show policy-map](#), on page 77
- [show processes](#), on page 77
- [show processes memory](#), on page 78
- [show rrm](#), on page 79
- [show rrm rogue containment](#), on page 80
- [show rrm rogue detection](#), on page 81
- [show running-config](#), on page 82
- [show security data-corruption](#), on page 83
- [show security system state](#), on page 83
- [show spectrum](#), on page 84
- [show tech-support](#), on page 85
- [show version](#), on page 85
- [show trace dot11\\_chn](#), on page 86
- [show trace](#), on page 86
- [show wips](#), on page 87

## show ap client-trace status

To view the AP client trace details, use the **show ap client-trace status** command.

```
show ap client-trace { events { all | mac word | system } | skb { drop-list | stats } | status }
```

### Syntax Description

<b>events</b>	View client trace event information
<b>all</b>	Displays all client trace events
<b>system</b>	Displays all system events
<b>mac</b>	Displays client trace events for specific MAC address
<i>word</i>	Specific client MAC address
<b>skb</b>	Displays client trace SKB information
<b>drop-list</b>	Displays client trace SKB drop list information
<b>stats</b>	Displays client trace SKB statistics
<b>status</b>	Displays client trace configuration

### Command Modes

Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the AP client trace status:

```
cisco-ap# show ap client-trace status
```

## show arp

To view the ARP table, use the **show arp** command.

**show arp**

Syntax Description	
	<b>arp</b> Shows ARP table

Command Modes	
	User EXEC (>) Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows a sample output of the command:

```
cisco-ap# show arp

Address Age (min)      Hardware Addr
  9.11.8.1             0 84:80:2D:A0:D2:E6
  9.11.32.111          0 3C:77:E6:02:33:3F
```

## show avc cft

To view the AVC client flow table information, use the **show avc cft** command.

**show avc cft word**

Syntax Description	
	<i>word</i> Client MAC address

Command Modes	
	User EXEC (>) Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

The following example shows how to view the AVC client flow table:

```
cisco-ap# show avc cft 02:35:2E:03:E0:F2
```

## show avc nbar

To view the AVC NBAR information, use the **show avc nbar** command.

```
show avc nbar {statistics | build | version}
```

Syntax Description	
<b>statistics</b>	Displays NBAR build details
<b>build</b>	Displays NBAR statistics
<b>version</b>	Displays NBAR and PP version

Command Modes	
	User EXEC (>)
	Privileged EXEC (#)

Command History	Release Modification
	8.1.111.0 This command was introduced.

The following example shows how to view the AVC NBAR build information:

```
cisco-ap# show avc nbar build
```

## show avc netflow flows

To list all the flows currently cached and to be sent to the Cisco WLC, use the **show avc netflow flows** command.

```
show avc netflow flows {download | upload}
```

Syntax Description	
<b>download</b>	Lists currently cached download flows
<b>upload</b>	Lists currently cached upload flows

Command Modes	
	User EXEC (>)
	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view all the currently cached flows:

```
cisco-ap# show avc netflow flows
```

## show avc status

To list the AVC provisioning status per WLAN/VAP, use the **show avc status** command.

**show avc status**

Command Modes	User EXEC (>)	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view AVC provisioning status per WLAN/VAP:

```
cisco-ap# show avc status

VAP FNF-STATUS AVC-QOS-STATUS
 0 Disabled Disabled
 1 Disabled Disabled
 2 Disabled Disabled
 3 Disabled Disabled
 4 Disabled Disabled
 5 Disabled Disabled
 6 Disabled Disabled
 7 Disabled Disabled
 8 Disabled Disabled
 9 Disabled Disabled
10 Disabled Disabled
11 Disabled Disabled
12 Disabled Disabled
13 Disabled Disabled
14 Disabled Disabled
15 Disabled Disabled
```

## show boot

To show boot attributes, use the **show boot** command.

**show boot**

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view boot attributes:

```
cisco-ap# show boot

BOOT path-list:      part2
Console Baudrate:   9600
Enable Break:       yes
Manual Boot:        no
Memory Debug:       no
Crashkernel:
```

## show capwap

To display CAPWAP options, use the **show capwap** command.

**show capwap** [{ip | mcast | traffic}]

<b>Syntax Description</b>	<b>client</b>	CAPWAP client information
	<b>ids</b>	CAPWAP ID information
	<b>ip</b>	CAPWAP IP configuration
	<b>location</b>	CAPWAP location information
	<b>mcast</b>	CAPWAP multicast information
	<b>pnp</b>	PNP information
	<b>traffic</b>	CAPWAP traffic information

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view the CAPWAP multicast information:

```
cisco-ap# show capwap mcast
```

## show capwap client

To display CAPWAP client information, use the **show capwap client** command.

```
show capwap client {callinfo info | detailrcb | rcb | config | ha | msginfo | timers | traffic}
```

Syntax Description	
<b>callinfo</b> <i>info</i>	CAPWAP client call information
<b>detailrcb</b>	CAPWAP client detailed RCB information
<b>rcb</b>	CAPWAP client RCB information
<b>config</b>	CAPWAP client config information
<b>ha</b>	CAPWAP client HA parameters
<b>msginfo</b>	CAPWAP client messages information
<b>timers</b>	CAPWAP client timers
<b>traffic</b>	CAPWAP client 802.11 traffic information

Command Modes	
	User EXEC (>)
	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view CAPWAP client traffic information:

```
cisco-ap# show capwap client traffic
```

## show capwap client trace

To display CAPWAP trace, use the **show capwap client trace** command.

```
show capwap client trace {clear | delete | disable | save | start | stop}
```

Syntax Description	
<b>clear</b>	Clears trace
<b>delete</b>	Deletes trace
<b>disable</b>	Disables trace at boot
<b>enable</b>	Enables trace at boot

---

<b>save</b>	Saves trace
-------------	-------------

---

<b>start</b>	Starts trace
--------------	--------------

---

<b>stop</b>	Stops trace
-------------	-------------

---

**Command Modes**

User EXEC (&gt;)

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0	This command was introduced.
-----------	------------------------------

---

The following example shows how to view CAPWAP client trace:

```
cisco-ap# show capwap client trace
```

## show capwap ids sig

To display CAPWAP ID signatures, use the **show capwap ids sig** command.

```
show capwap ids sig [{list | stats}]
```

**Syntax Description**


---

<b>list</b>	Signature list entries
-------------	------------------------

---

<b>stats</b>	Signature attack statistics
--------------	-----------------------------

---

**Command Modes**

User EXEC (&gt;)

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0	This command was introduced.
-----------	------------------------------

---

The following example show how to view CAPWAP ID signature statistics:

```
cisco-ap# show capwap ids sig stats
```

## show cdp

To display CDP options, use the **show cdp** command.

```
show cdp {entry device device-name | inline_power | interface | neighbors | traffic}
```



<b>Syntax Description</b>	<b>entry device</b> <i>device-name</i> Information for specific neighbor entry whose name you must enter
	<b>inline_power</b> Inline power negotiation information
	<b>interface</b> CDP interface status and configuration
	<b>neighbors</b> CDP neighbor entries
	<b>traffic</b> CDP statistics

**Command Modes** Privileged EXEC (#)

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

The following example shows how to view information for a specific neighbor entry:

```
cisco-ap# show cdp entry device mydevice
```

## show class-map

To display CPL class map, use the **show class-map** command.

**show class-map**

**Command Modes** User EXEC (>)  
Privileged EXEC (#)

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

The following example shows how to view CPL class map:

```
cisco-ap# show class-map
```

## show cleanair debug

To display cleanair debug settings, use the **show cleanair debug** command.

**show cleanair debug**

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view CleanAir debug settings:

```
cisco-ap# show cleanair debug
```

## show client statistics

To display client statistics, use the **show client statistics** command.

**show client statistics** *client-mac-address*

Syntax Description	
	<i>client-mac-address</i> MAC address of the client

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view client statistics:

```
cisco-ap# show client statistics 70:DB:98:66:34:FA
```

## show clock

To display the system clock, use the **show clock** command.

**show clock**

**Command Modes** User EXEC (>)  
Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the system clock:

```
cisco-ap# show clock
```

# show configuration

To display the contents of the non-volatile memory, use the **show configuration** command.

## show configuration rlan

<b>Command Modes</b>	Privileged EXEC (#)
<b>Syntax Description</b>	<b>rlan</b> Displays the RLAN configuration.

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	8.1.111.0	This command was introduced.
	8.9	This command was enhanced by adding <b>rlan</b> parameter.
	8.10.112.0	The output of this command was enhanced to show the status of broken antenna detection.

The following example shows how to view the AP configuration details:

```
cisco-ap# show configuration

AP Name                : AP58AC.78DC.C2F0
Admin State            : Enabled
AP Mode                : FlexConnect
AP Submode             : Not Configured
Location               : default location
Reboot Reason          : Reload command
.
.
AP Link LAG status     : Disabled
AP WSA Mode            : Enabled
Vlan Interface         : Disabled

Broken antenna detection : Enabled (Global)
RSSI Failure Threshold : 40
Weak RSSI               : 60
Detection Time          : 12
If any broken antenna? : ALL
AP58AC.78DC.C2F0#
```

# show controller ble

To view Bluetooth Low Energy radio interface parameter information, use the **show controller ble** command.

```
show controller ble ble-interface-number {broadcast | counters | floor-tag floor-beacon-mac-addr | interface | local | scan {brief | detail floor-beacon-mac-addr} | timers}
```

<b>Syntax Description</b>	<i>ble-interface-number</i>	BLE interface number that you must enter; Valid value is 0
	<b>broadcast</b>	Displays BLE broadcast summary information

<b>counters</b>	Displays BLE transport counters information
<b>floor-tag</b> <i>floor-beacon-mac-addr</i>	Displays sync data of the floor beacon whose MAC address you must specify
<b>interface</b>	Displays BLE interface summary information
<b>local</b>	Displays sync information of host BLE radio
<b>scan brief</b>	Displays brief BLE scan summary information
<b>scan detail</b> <i>floor-beacon-mac-addr</i>	Displays BLE scan summary information in detail; you must specify the floor beacon MAC address
<b>timers</b>	Displays BLE timers information

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.7 This command was introduced.

**Examples**

To view the BLE timers information, use this command:

```
cisco-ap# show controller ble 0 timers

Timers
-----
Scan timer status      : Running
Scan timer interval    : 10 secs
Scan started at       : 0D:00H:04M:28S ago
Last scan done at     : 0D:00H:00M:06S ago
```

If scanning is working as expected, the 'Last scan done at' time should always be less than or equal to the scan interval set.

## show controllers dot11Radio

To display dot11 interface information, use the **show controllers dot11Radio** command.

```
show controllers dot11Radio dot11-interface-no{antenna | { atfconfiguration | statistics } | bandselect
| client { client-mac-addr | all detail } | frequency | powercfg | powerreg | radiostats | rate | vlan
| wlan { wlan-id | all detail } }
```

**Syntax Description**

<i>dot11-interface-no</i>	Dot11Radio interface number.
<b>atf configuration</b>	Displays the AirTime Fairness configuration.
<b>atf statistics</b>	Displays the AirTime Fairness statistics.

<b>bandselect</b>	Displays the bandselect statistics.
<b>antenna</b>	Displays the antenna settings
<b>client</b> <i>client-mac-addr</i>	Displays the details of the client whose MAC address is specified.
<b>detail</b>	Displays the TID statistics for all the clients.
<b>frequency</b>	Displays the frequency information.
<b>powercfg</b>	Displays the configured power information.
<b>powerreg</b>	Displays the transmit power information.
<b>radio-stats</b>	Displays the radio statistics.
<b>rate</b>	Displays the rate information.
<b>vlan</b>	Displays the VLAN summary.
<b>wlan</b> <i>wlan-id</i>	Displays the VLAN/WLAN details of the WLAN ID specified.
<b>detail</b>	Displays the TID statistics for all the clients.

**Command Modes** User EXEC (>)

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

8.9 This command was enhanced by adding the **bandselect** , **client all detail** , **wlan** parameters.

The following example shows how to view 802.11 interface information for interface number 1:

```
cisco-ap# show controllers dot11Radio 1
```

## show controllers nss status

To display NSS information, use the **show controllers nss status** command.

**show controllers nss status**

**Command Modes** User EXEC (>)

Privileged EXEC (#)

**Command History**

**Release Modification**

8.1.111.0 This command was introduced.

The following example shows how to view NSS information:

```
cisco-ap# show controllers nss status
```

## show controllers wired

To view the wired interface, use the **show controllers wired** command.

**show controllers wired** *wired-interface-number*

<b>Syntax Description</b>	<i>wired-interface-number</i> Wired interface number from 0 to 3				
<b>Command Modes</b>	Privileged EXEC (#)				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>8.1.111.0</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	8.1.111.0	This command was introduced.
Release	Modification				
8.1.111.0	This command was introduced.				

The following example shows how to view information about the controllers' wired interface whose ID is 1:

```
cisco-ap# show controllers wired 1

wired1    Link encap:Ethernet  HWaddr C8:8B:6A:33:59  eMac Status: DOWN
          inet addr:9.11.8.104  Bcast:9.255.255.255  Mask:255.255.255.255
          DOWN BROADCAST RUNNING PROMISC MULTICAST  MTU:2400  Metric:1
          RX packets:38600 errors:0 dropped:1 overruns:0 frame:0
          TX packets:179018 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:80
          RX bytes:3812643 (3.6 MiB)  TX bytes:54721869 (52.1 MiB)

Gig Emacl Counters
-----
0 Good octets rx, 0 Bad octets rx, 0 Unicast frames rx,
0 Broadcast frames rx, 0 Multicast frames rx, 0 64 byte frames rx,
0 65_TO_127 byte frames, 0 128_TO_255 byte frames, 0 256_TO_511 byte frames,
0 512_TO_1023 byte frames, 0 1024_TO_MAX byte frames, 0 Good octets tx,
0 Unicast frames tx, 0 Multicast frames tx, 0 Broadcast frames tx,
0 Crc errors sent, 0 Flow control rx, 0 Flow control tx,
0 Rx fifo overrun, 0 Undersized rx, 0 Fragments rx,
0 Oversize rx, 0 Jabber rx, 0 Mac rx error,
0 Bad crc event, 0 Collision, 0 Late collision,
```

## show crypto

To view the crypto attributes, use the **show crypto** command.

**show crypto**

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view the crypto attributes:

```
cisco-ap# show crypto
```

## show debug

To view the debugs enabled, use the **show debug** command.

```
show debug
```

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view the debugs that are in enabled state:

```
cisco-ap# show debug
```

## show dhcp

To view the status of Dynamic Host Configuration Protocol (DHCP), use the **show dhcp** command.

```
show dhcp {lease | servers}
```

<b>Syntax Description</b>	<b>lease</b>	Displays the DHCP addresses leased from a server
	<b>servers</b>	Displays the known DHCP servers

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

The following example shows how to view the status of DHCP addresses leased from a server:

```
cisco-ap# show dhcp lease
```

## show dot11 qos

To view the Quality of Service (QoS) parameters for 802.11 network, use the **show dot11 qos** command.

```
show dot11 qos
```

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0 This command was introduced.

---

The following example shows how to view the Quality of Service (QoS) parameters for 802.11 network:

```
cisco-ap# show dot11 qos
```

## show dot11 wlan wpa3

To view the WPA3 configuration on an 802.11 network, use the **show dot11 wlan wpa3** command.

```
show dot11 wlan wpa3 [transition]
```

**Syntax Description**

<b>transition</b>	Shows details of WPA3 transition mode.
-------------------	--

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.10 This command was introduced.

---

The following example shows how to view the WPA3 configuration on an 802.11 network:

```
cisco-ap# show dot11 wlan wpa3
```



# show filesystems

To view the filesystem information, use the **show filesystems** command.

**show filesystems**

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view the filesystem information:

```
cisco-ap# show filesystems
```

```
Filesystem          Size      Used Available Use% Mounted on
/dev/ubivol/storage 57.5M     1.9M     52.6M     4% /storage
```

# show flash

To view the flash contents, use the **show flash** command.

**show flash** [{**cores** [**detail** *core-file-name* ] | **crash** | **syslogs**}]

<b>Syntax Description</b>	<b>cores</b>	Displays the core files in flash
	<b>detail</b>	Displays the core file contents
	<i>core-file-name</i>	The core file name
	<b>crash</b>	Displays the crash files in flash
	<b>syslogs</b>	Displays the syslogs files in flash

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

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

The following example shows how to view the details of a core file in flash:

```
cisco-ap# show flash cores detail filename1
```

## show flexconnect

To view the flexconnect information for an access point, use the **show flexconnect** command.

```
show flexconnect {calea | cckm | client [aaa-override | counter | priority] | dot11r |
mcast | oeap | pmk | status | vlan-acl | wlan }
```

Syntax	Description
<b>calea</b>	Displays the calea information
<b>cckm</b>	Displays the CCKM cache entry information
<b>client</b>	Displays the client information
<b>aaa-override</b>	Specifies the AAA override parameters
<b>counter</b>	Specifies the counter for all clients
<b>priority</b>	Specifies the client priority
<b>dot11r</b>	Displays the 802.11r cache entry information
<b>mcast</b>	Displays the multicast information
<b>oeap</b>	Displays the FlexConnect OEAP information
<b>pmk</b>	Displays the OKC or PMK cache entry information
<b>status</b>	Displays the standalone status
<b>vlan-acl</b>	Displays the VLAN ACL mapping
<b>wlan</b>	Displays the WLAN configuration

Command Modes
User EXEC (>)
Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the information about a client of a FlexConnect AP:

```
cisco-ap# show flexconnect client
```

## show flexconnect oeap firewall

To view the OEAP firewall information, use the **show flexconnect oeap firewall** command.

```
show flexconnect oeap firewall [{dmz | filtering | forwarding}]
```

Syntax Description	dmz	Displays the OEAP firewall DMZ information
	filtering	Displays the OEAP firewall filtering information
	forwarding	Displays the OEAP firewall port forwarding information

Command Modes	User EXEC (>) Privileged EXEC (#)
---------------	--------------------------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the OEAP firewall DMZ information:

```
cisco-ap# show flexconnect oeap firewall dmz
```

## show flexconnect wlan

To view the WLAN configuration for Flexconnect AP mode, use the **show flexconnect wlan** command.

```
show flexconnect wlan [{l2acl | qos | vlan}]
```

Syntax Description	l2acl	Specifies the Layer 2 ACL mapping for WLAN
	qos	Specifies the QoS parameters for WLAN
	vlan	Specifies the VLAN mapping for WLAN

Command Modes	User EXEC (>) Privileged EXEC (#)
---------------	--------------------------------------

Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the WLAN Layer 2 ACL mapping for the Flexconnect AP:

```
cisco-ap# show flexconnect wlan l2acl
```

## show interfaces dot11Radio

To view the interface status and configuration for an 802.11 radio, use the **show interfaces dot11Radio** command.

```
show interfaces dot11Radio radio-interface-number {dfs | memory [memory-address length | firmware] | mumimo wlan-number | sniffer | statistics | wlanwlan-id datapathcounters | statistics }
```

### Syntax Description

<i>radio-interface-number</i>	Specifies the interface number for 802.11 radio. The valid range is from 0 to 1
<b>dfs</b>	Displays the DFS statistics
<b>memory</b>	Displays the dump radio memory
<i>memory-address</i>	Specifies the memory address. The valid range is between 0 and ffffffff
<i>length</i>	Specifies the length. The valid range is from 0 to 64
<b>firmware</b>	Dumps firmware logs
<b>mumimo</b>	Displays the multiuser MIMO statistics information
<i>wlan-number</i>	The 802.11-specific value whose valid range is from 0 to 15.
<b>sniffer</b>	Displays the sniffer mode statistics
<b>statistics</b>	Displays the statistics information for 802.11 radio <b>Note</b> Cisco 1852, 9117, 9130 APs do not include the beacon tx statistics under the 802.11 tx statistics counter.
<b>wlan</b> <i>wlan-id</i>	Displays the specified WLAN information
<b>datapath</b>	Displays the datapath counters.
<b>counters</b>	Displays the datapath counters and drops.

### Command Modes

Privileged EXEC (#)

### Command History

#### Release Modification

8.1.111.0 This command was introduced.

8.9 This command was enhanced by adding the **datapath** parameter.

The following example shows how to view the DFS statistics for a 802.11 interface whose number is 1:

```
cisco-ap# show interfaces dot11Radio 1 dfs
```

```
DFS Data:
```

```
Radar Detected:          0
Inactive Radar Detected: 0
```

## show interfaces network

To view the Linux network interfaces, use the **show interfaces network** command.

**show interfaces network**

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

The following example shows how to view the Linux network interfaces:

```
cisco-ap# show interfaces network
```

## show interfaces wired

To view the wired interface, use the **show interfaces wired** command.

**show interfaces wired** *wired-interface-number* {**MIB-stats** | **datapath counters**}

<b>Syntax Description</b>	<i>wired-interface-number</i>	Wired interface number; valid range is between 0 to 3
	<b>MIB-stats</b>	Displays the AP internal-Switch MIB counters.
	<b>datapath</b>	Displays the datapath counters.
	<b>counters</b>	Displays the datapath counters and drops.

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	8.1.111.0	This command was introduced.
	8.9	This command was enhanced by adding the <b>datapath</b> parameter.

The following example shows how to view the wired interface whose number is 1:

```
cisco-ap# show interfaces wired 1
```

# show inventory

To view the physical inventory, use the **show inventory** command.

## show inventory

### Command Modes

User EXEC (>)

Privileged EXEC (#)

### Command History

Release	Modification
8.1.111.0	This command was introduced.

The following example shows how to view the physical inventory:

```
cisco-ap# show inventory
```

```
NAME: AP2800, DESCR: Cisco Aironet 2800 Series (IEEE 802.11ac) Access Point
PID: AIR-AP2802I-D-K9 , VID: V01, SN: XXXXXXXXXXXX
```

# show ip

To view the IP information, use the **show ip** command.

```
show ip {access-lists | interface brief | route | tunnel [eogre {domain | forwarding-table | gateway} | fabric | summary | sip-snooping {stats | status} ] }
```

### Syntax Description

<b>access-lists</b>	Lists the IP access lists
<b>interface</b>	Displays the IP interface status and configuration
<b>brief</b>	Displays the brief summary of IP status and configuration
<b>route</b>	Displays the IP routing table
<b>tunnel</b>	Displays the IP tunnel information
<b>eogre</b>	Displays the EoGRE tunnel information
<b>domain</b>	Displays the EoGRE tunnel domain information
<b>forwarding-table</b>	Displays the EoGRE tunnel encapsulation and decapsulation information
<b>gateway</b>	Displays the EoGRE tunnel gateway information
<b>fabric</b>	Displays the IP fabric tunnel information
<b>summary</b>	Displays the information for all tunnels

<b>sip-snooping</b>	Displays the SIP snooping options.
<b>stats</b>	Displays the transmitted and received SIP snooping statistics.
<b>status</b>	Displays the SIP snooping status.

<b>Command Modes</b>	User EXEC (>)
	Privileged EXEC (#)

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	8.1.111.0	This command was introduced.
	8.9	This command was enhanced by adding the <b>sip-snooping</b> parameter.

The following example shows how to view information about the lists the IP access lists:

```
cisco-ap# show ip access-lists
```

## show lacp

To view the Link Aggregation Control Protocol (LACP) options, use the **show lacp** command.

```
show lacp {counters | internal | neighbors}
```

<b>Syntax Description</b>	<b>counters</b>	Displays traffic information
	<b>internal</b>	Displays internal information
	<b>neighbors</b>	Displays LACP neighbor entries

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

The following example shows how to view the LACP traffic information:

```
cisco-ap# show lacp counters
```

## show logging

To view the contents of logging buffers, use the **show logging** command.

```
show logging
```

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

The following example shows how to view the contents of logging buffers:

```
cisco-ap# show logging
```

## show memory

To display memory usage on an access point, use the **show memory** command.

```
show memory [{detail | pool | summary}]
```

<b>Syntax Description</b>		
<b>detail</b>	Displays	detailed system memory usage
<b>pool</b>	Displays	system memory pool
<b>summary</b>	Display	system memory usage statistics

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

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

The following example shows how to view the system memory usage statistics:

```
cisco-ap# show memory
Memory summary:
MemTotal:          1030608 kB
MemFree:           713832 kB
MemAvailable:      710492 kB
Buffers:           0 kB
Cached:            88224 kB
SwapCached:        0 kB
Active:            28932 kB
Inactive:          82872 kB
Active(anon):      28900 kB
Inactive(anon):    82812 kB
Active(file):       32 kB
Inactive(file):    60 kB
Unevictable:       0 kB
Mlocked:           0 kB
SwapTotal:         0 kB
SwapFree:          0 kB
Dirty:             0 kB
Writeback:         0 kB
AnonPages:         23580 kB
Mapped:            11380 kB
```



```

Shmem:          88132 kB
Slab:           132140 kB
SReclaimable:   3368 kB
SUnreclaim:    128772 kB
KernelStack:   864 kB
PageTables:    748 kB
NFS_Unstable:   0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:   515304 kB
Committed_AS:  193960 kB
VmallocTotal:  1024000 kB
VmallocUsed:   69808 kB
VmallocChunk:  915324 kB

```

```

System Memory:
              total      used      free      shared      buffers
Mem:         1030608    316848    713760         0         0
-/+ buffers:         316848    713760
Swap:        0         0         0

```

## show policy-map

To view policy maps on access point, use the **show policy-map** command.

### show policy-map

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

<b>Command History</b>	<b>Release</b> <b>Modification</b>
------------------------	------------------------------------

8.1.111.0	This command was introduced.
-----------	------------------------------

The following example shows how to view the policy maps on the access point:

```
cisco-apshow policy-map
```

## show processes

To view process utilization details, use the **show processes** command.

```
showprocesses {cpu cpu-number | dmalloc {capwap | wcp} | status}
```

<b>Syntax Description</b>	<b>cpu</b> <i>cpu-number</i> Displays the specified CPU's utilization of the processes; valid range of values for the CPU number is between 0 to 3
	<b>dmalloc</b> Displays the process utilization of the dmalloc processes
	<b>capwap</b> Displays dmalloc statistics for CAPWAP
	<b>wcp</b> Displays dmalloc statistics for WCP

---

<b>status</b>	Displays watchdog process status
---------------	----------------------------------

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0	This command was introduced.
-----------	------------------------------

---

The following example shows how to view the process watchdog status:

```
cisco-ap# show processes status
      Process                Alive      Monitored
      capwap                 True       True
      switchdrv             True       False
      wcpd                   True       True
      kclic                  True       True
      cleanaird              True       True
      mrvlfd                 True       True
```

## show processes memory

To display the processes on the access point, use the **show processes memory** command.

**show processes memory** {maps | smaps} pid *pid-number*

**Syntax Description**


---

<b>maps</b>	Displays maps for the processes
<b>smaps</b>	Displays smaps for the processes
<b>pid</b>	Process ID that you have to specify <i>pid-number</i>

---

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**


---

8.1.111.0	This command was introduced.
-----------	------------------------------

---

The following example shows how to view the list of processes utilizing the memory on the access point:

```
cisco-ap# show processes memory

Mem total:1030608 anon:23876 map:11424 free:712728
  slab:132748 buf:0 cache:88284 dirty:0 write:0
Swap total:0 free:0
  PID  VSZ^VSZRW  RSS (SHR) DIRTY (SHR) STACK COMMAND
  6227 56500 53464 1168 732 1144 732 132 /usr/sbin/mrvlfd
  6283 27536 20668 13032 2400 13032 2400 132 /usr/sbin/capwapd
  6297 24880 10612 14536 1376 14536 1376 132 wcpd
```

```

6255 9612 6600 1508 1052 1508 1052 132 /usr/sbin/cleanaird
5122 9556 4144 2664 2012 2664 2012 132 /usr/bin/capwap_brain
29097 7148 1536 3560 2392 3556 2388 132 /usr/sbin/cisco_shell
3142 6828 1216 2992 2264 2992 2264 132 /usr/sbin/cisco_shell
5106 4588 404 1912 1644 1912 1644 132 /usr/bin/fastcgi -s /tmp/fcgi_sock
5108 4588 404 1912 1644 1912 1644 132 /usr/bin/slowcgi -s /tmp/slow_fcgi_sock
6084 4544 452 928 360 928 360 132 /usr/sbin/lighttpd -f /etc/lighttpd.conf
6214 3692 344 1420 960 1420 960 132 tamd_proc ap-tam 1 0 -debug err
6213 3556 340 1460 1104 1460 1104 132 tams_proc -debug err
6133 3396 400 1196 976 1196 976 132 /usr/bin/poder_agent
4689 3176 336 1012 812 1012 812 132 /usr/bin/sync_log /storage/syslogs/13
6143 3140 304 1428 1204 1428 1204 132 /usr/bin/failover
4716 3136 284 616 436 616 436 132 watchdogd
6121 3116 280 988 820 988 820 132 bigacl_d
5084 3112 272 952 804 952 804 132 /usr/bin/led_core
6181 1884 320 1044 260 1044 260 132 perl /usr/bin/drt.pl
1 1596 196 492 412 492 412 132 init
30914 1596 196 428 344 428 344 132 top -m -b -n 1
6145 1596 196 248 176 248 176 132 {S80cisco} /bin/sh /etc/init.d/S80cisco
start
30912 1592 192 424 356 424 356 132 {show_process_me} /bin/ash
/usr/bin/cli_scripts/show_process_memory.sh 0 0 0 0 0 0 0 0 0
30911 1592 192 400 336 400 336 132 /bin/sh -c
/usr/bin/cli_scripts/show_process_memory.sh 0 0 0 0 0 0 0 0 | more
4684 1592 192 368 304 368 304 132 syslogd -S -s 100 -b 1 -L -R 255.255.255.255
30913 1592 192 332 264 332 264 132 more
4688 1584 184 344 284 344 284 132 klogd
4686 1584 184 320 264 320 264 132 printkd
30906 1584 184 284 228 284 228 132 sleep 10
29085 1452 332 640 416 640 416 132 /usr/sbin/dropbear -E -j -k -d
/storage/dropbear/dropbear_dss_host_key -r /storage/dropbear/dropbear_rsa_host_key
6209 1384 264 416 364 416 364 132 /usr/sbin/dropbear -E -j -k -d
/storage/dropbear/dropbear_dss_host_key -r /storage/dropbear/dropbear_rsa_host_key
8411 1096 212 444 336 444 336 132 dnsmasq -C /etc/dnsmasq.host.conf
6115 1096 212 436 340 436 340 132 dnsmasq -C /etc/dnsmasq.vaperr.conf

```

## show rrm

To view the Radio Resource Management (RRM) properties, use the **show rrm** command.

```
show rrm {hyperlocation [level-list] | neighbor-list [details] | receive {configuration | statistics}}
```

Syntax Description	
<b>hyperlocation</b> <i>level-list</i>	Displays status of Cisco Hyperlocation on the AP
<b>neighbor-list</b>	Displays neighbor-list statistics
<b>receive</b>	Receive signal strength indicator (RSSI) of the AP
<b>rogue</b>	Displays rogue-related information

**Command Modes** Privileged EXEC (#)

**Command History** **Release** **Modification**

8.1.111.0 This command was introduced.

**Usage Guidelines**

The following example shows how to view the level 1 channel scan list in Hyperlocation:

```
cisco-ap# show rrm hyperlocation level1-list
Level-1 List for 2.4GHz Band
=====
Channel   Width      Serving MAC   Max Clients
-----
Level-1 List for 5GHz Band
=====
Channel   Width      Serving MAC   Max Clients
-----
```

## show rrm rogue containment

To view rogue containment information on an access point, use the **show rrm rogue containment** command.

```
show rrm rogue containment {ignore | info} Dot11Radio radio-interface-number
```

**Syntax Description**

<b>ignore</b>	Displays list of rogue APs that are configured to be ignored
<b>info</b>	Displays rogue containment configuration and statistics for an AP
<b>Dot11Radio</b>	Specifies the <b>Dot11Radio</b> interface keyword.
<i>radio-interface-number</i>	Slot of the radio interface; valid values are 0 and 1

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

The following example shows how to view the rogue containment and statistics for the 802.11 interface numbered 1:

```
cisco-ap# show rrm rogue containment info Dot11Radio 1
Rogue Containment Info and Stats for slot 1:
bssid client-addr contain-type channels

Request Status count
      Submit      0
      Success     0
      Timeout     0
      Error       0
      Tuned       0
      Flushed     0
      Bad Channel  0
      Tail Dropped 0
      Cancelled   0
NDP DFS Tx Cancelled 0
      Tx Failed   0
      Created     0
```

# show rrm rogue detection

To view RRM rogue detection configuration parameters, use the **show rrm rogue detection** command.

```
show rrm rogue detection {adhoc | ap | clients | config | rx-stats} Dot11Radio radio-interface-number
```

Syntax Description	Parameter	Description
	<b>adhoc</b>	Displays the primary ad hoc rogue AP list for a 802.11 radio slot; valid values are 0 and 1
	<b>ap</b>	Displays rogue detection parameters for the AP for a 802.11 radio slot; valid values are 0 and 1
	<b>clients</b>	Displays primary list of rogue clients
	<b>config</b>	Displays rogue detection configuration on the AP
	<b>rx-stats</b>	Displays rogue detection receive statistics on the 802.11 interfaces of an AP
	<b>Dot11Radio</b>	Specifies 802.11 radio interface
	<i>radio-interface-number</i>	The 802.11 radio interface number; valid values are 0 and 1

**Command Modes** Privileged EXEC (#)

**Command History** **Release Modification**

8.1.111.0 This command was introduced.

The following example shows how to view the RRM rogue detection configuration details:

```
cisco-ap# show rrm rogue detection config

Rogue Detection Configuration for Slot 0:
Rogue Detection Mode : Enabled
Rogue Detection Report Interval : 10
Rogue Detection Minimum Rssi : -90
Rogue Detection Transient Interval : 0
Rogue Detection Flex Contain : Disabled
Rogue Detection Flex Contain Adhoc : Disabled
Rogue Detection Flex Contain SSID : Disabled
Rogue Containment Autorate : Disabled
Scan Duration : 180000
Channel Count : 11
Transient Threshold : 0

Rogue Detection Configuration for Slot 1:
Rogue Detection Mode : Enabled
Rogue Detection Report Interval : 10
Rogue Detection Minimum Rssi : -90
Rogue Detection Transient Interval : 0
Rogue Detection Flex Contain : Disabled
Rogue Detection Flex Contain Adhoc : Disabled
Rogue Detection Flex Contain SSID : Disabled
Rogue Containment Autorate : Disabled
```

```
Scan Duration : 180000
Channel Count : 25
Transient Threshold : 0
```

## show running-config

To display the contents of the currently running configuration on the access point, use the **show running-config** command.

### show running-config

#### Command Modes

Privileged EXEC (#)

#### Command History

##### Release Modification

8.1.111.0 This command was introduced.

The following example shows how to view the contents of the currently running configuration on the access point:

```
cisco-ap# show running-config

AP Name                : ap1540
Admin State            : Enabled
AP Mode                : Local
AP Submode             : None
Location               : default location
Reboot Reason          : Config Mwar
Primary controller name : cisco_3504
Primary controller IP  : <controller-ip-address>
Secondary controller name :
Secondary controller IP :
Tertiary controller name :
Tertiary controller IP  :
Controller from DHCP offer : <controller-dhcp-server-address>
Controller from DNS server : <controller-dns-server-address>
AP join priority       : 1
IP Prefer-mode         : IPv4
CAPWAP UDP-Lite       : Unconfigured
Last Joined Controller name: wlc3504
DTLS Encryption State  : Disabled
Discovery Timer        : 10
Heartbeat Timer        : 30
CDP State              : Enabled
Watchdog monitoring    : Enabled
IOX                    : Disabled
RRM State              : Enabled
LSC State              : Disabled
SSH State              : Enabled
AP Username            : admin
Session Timeout        : 0
Extlog Host            : 0.0.0.0
Extlog Flags           : 0
Extlog Status Interval : 0
Syslog Host            : <syslog-host-ip-address>
Syslog Facility        : 0
```

```

Syslog Level           : errors
Core Dump TFTP IP Addr :
Core Dump File Compression : Disabled
Core Dump Filename    :
Client Trace Status   : Enabled(All)
Client Trace All Clients : Enabled
Client Trace Filter    : 0x0000000E
Client Trace Out ConsoleLog: Disabled
WLC Link LAG status   : Disabled
AP Link LAG status    : Disabled
AP WSA Mode           : Disabled

```

## show security data-corruption

To view data inconsistency errors, use the **show security data-corruption** command.

### show security data-corruption

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

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.7	This command was introduced.

### Examples

The following example shows how to view data inconsistency errors:

```
cisco-ap# show security data-corruption
```

## show security system state

To view the current state of system-level security, use the **show security system state** command.

### show security system state

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

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.7	This command was introduced.

## Examples

To view the current state of system-level security, use this command:

```
cisco-ap# show security system state

XSPACE:
    Non-Executable stack:    Yes
    Non-Executable heap:    Yes
    Non-Writable text:      Yes

OSC:
    Version:                 1.1.0

SafeC:
    Version:                 3.1.1
```

The table below describes the significant fields shown in the display:

**Table 4: show security system state Field Descriptions**

Field	Description
Non-Executable stack	Indicates whether the system prevents execution from the stack
Non-Executable heap	Indicates whether the system prevents execution from the heap
Non-Writable text	Indicates whether the system prevents the text section from being writable
OSC version	Indicates the version of the OSC library used by the applications
SafeC version	Indicates the version of the SafeC library used by the applications

## show spectrum

To view the show commands of the spectrum firmware, use the **show spectrum** command.

```
show spectrum {list | recover | status }
```

Syntax Description	
<b>list</b>	Lists the spectrum FW data files
<b>recover</b>	Displays the spectrum FW recover count
<b>status</b>	Displays the spectrum FW status
Command Modes	Privileged EXEC (#)



Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to view the spectrum firmware status:

```
cisco-ap# show spectrum status

Spectrum FW status slot 0:
  version: 1.15.4
  status:  up, crashes 0, resets 0, radio reloads 0
  load:    37.00 34.75 33.50 33.25
  NSI Key: 26c1bd25893a4b6dd3a00fe71735d067
  NSI:     not configured
  reg_wdog: 255 26309 0
  dfs_wdog: 0
  dfs_freq: 0
Spectrum FW status slot 1:
  version: 1.15.4
  status:  up, crashes 0, resets 0, radio reloads 0
  load:    37.25 38.00 38.75 39.00
  NSI Key: 26c1bd25893a4b6dd3a00fe71735d067
  NSI:     not configured
  reg_wdog: 255 26309 0
  dfs_wdog: 0
  dfs_freq: 0
```

## show tech-support

To automatically run show commands that display system information, use the **show tech-support** command.

**show tech-support**

Command Modes	Privileged EXEC (#)	
Command History	Release	Modification
	8.1.111.0	This command was introduced.

The following example shows how to automatically run show commands that display system information:

```
cisco-ap# show tech-support
```

## show version

To view the software version information of the AP, use the **show version** command.

**show version**

---

**Command Modes** Privileged EXEC (#)

---

Command History	Release	Modification
	8.1.111.0	This command was introduced.

---

The following example shows how to view the software version information of the AP:

```
cisco-ap# show version
```

## show trace dot11\_chn

To view off-channel events on 802.11 channel of an AP, use the **show trace dot11\_chn** command.

```
show trace dot11_chn {enable | disable | statistics}
```

Syntax Description	enable	disable	statistics
	Enables displaying of off-channel events on the 802.11 radio 0 and 1	Disables displaying of off-channel events on the 802.11 radios 0 and 1	Displays off-channel event statistics on 802.11 radios 0 and 1

---



---

**Command Modes** Privileged EXEC (#)

---

Command History	Release	Modification
	8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to view off-channel event statistics on 802.11 radios:

```
cisco-ap# show trace dot11_chn statistics

Dot11Radio0 Off-Channel Statistics:
total_count in_prog_count last-chan last-type last-dur
          0             0           0           0           0

Dot11Radio1 Off-Channel Statistics:
total_count in_prog_count last-chan last-type last-dur
          0             0           0           0           0
```

## show trace

To view trace logs on the AP, use the **show trace** command.

```
show trace
```

---

**Command Modes** Privileged EXEC (#)

---

**Command History** **Release** **Modification**


---

8.1.111.0 This command was introduced.

---

The following example shows how to view the trace logs on the AP:

```
cisco-ap# show trace
```

## show wips

To view details of the AP that is configured in wIPS mode, use the **show wips** command.

```
show wips {alarm alarm-id | analyzer | buffer | channel channelno | infrastructure-device | neighbors | node mac mac-address | node number number | object | policy policy-id | policy ssid | session mac-address | stats | violation node mac-address | violation channel channel-number}
```

Syntax Description		
<b>alarm</b>		Displays statistics of the configured alarm if the AP is configured in wIPS mode; valid values are between 0 and 255
<i>alarm-id</i>		Alarm ID; valid values are between 0 and 255
<b>analyzer</b>		Displays analyzer related statistics
<b>buffer</b>		Displays statistics of the buffer
<b>channel</b>		Displays channel related statistics
<i>channelno</i>		Channel number; valid values are between 0 and 255
<b>infrastructure-device</b>		Displays AP infrastructure information
<b>neighbors</b>		Displays statistics of neighbors.
<b>node</b>		Displays AP node information
<b>mac</b> <i>mac-address</i>		MAC address of the node.
<b>node</b>		Node.
<b>number</b> <i>number</i>		Node number; valid values are between 1 and 500
<b>object</b>		AP object store
<b>policy</b> { <i>policy-id</i>   <b>ssid</b> }		AP policy; you must specify either a policy ID or the policy SSID.
<b>session</b> <i>mac-address</i>		Displays node session details; you must enter the MAC address of the node

<b>stats</b>	Displays AP statistics
<b>violation</b>	Tracks AP violations
<b>node</b> <i>mac-address</i>	Tracks node-based violations
<b>channel</b> <i>channel-number alarm-id</i>	Tracks channel-based violations; you must enter channel number and alarm ID

**Command Modes**

Privileged EXEC (#)

**Command History****Release Modification**

8.1.111.0 This command was introduced.

The following example shows how to view the wIPS statistics information on the AP:

```
cisco-ap# show wips stats
```



## CHAPTER 8

# System Management Commands

- [ap-type](#) , on page 89
- [archive](#), on page 90
- [copy](#) , on page 90
- [delete](#), on page 91
- [disable](#), on page 92
- [enable](#), on page 92
- [exec-timeout](#) , on page 92
- [logging](#), on page 93
- [more](#), on page 93
- [reload](#), on page 94
- [terminal](#), on page 95

## ap-type

To configure the AP type for an AP, use the **ap-type** command.

```
ap-type {capwap | mobility-express word | workgroup-bridge}
```

<b>Syntax Description</b>	<b>capwap</b> Enable the AP as CAPWAP AP type
	<b>mobility-express</b> Enable the AP as Mobility Express AP type
	<i>word</i> Enter the TFTP transfer command details in following format: tftp://<tftp-server-ip-address>/<filename with path from root>
	<b>workgroup-bridge</b> Enable the Workgroup Bridge(WGB) AP type
<b>Command Modes</b>	Privileged EXEC (#)
<b>Command History</b>	<b>Release Modification</b>
	8.1.111.0 This command was introduced.
	8.8.120.0 This command was enhanced by added <b>workgroup-bridge</b> parameter.

### Examples

The following example shows how to configure the AP type to CAPWAP:

```
cisco-ap# ap-type capwap
```

## archive

To download the AP image, use the **archive** command.

```
archive download-sw {/no-reload | /reload | capwap word}
```

Syntax Description	
<b>download-sw</b>	Software download commands
<b>/no-reload</b>	No-reload after loading the image
<b>/reload</b>	Reload after loading the image
<b>capwap</b>	Download the image from the Cisco WLC
<i>word</i>	Enter the image details in the <b>ap image type ap3g3/ap1g4</b> format

Command Modes	
	Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

## copy

To copy a file, use the **copy** command.

```
copy {cores filename [scp: scp-url | tftp: tftp-url] | flash filename [scp: scp-url | tftp: tftp-url] | support-bundle [scp: scp-url | tftp: tftp-url] | syslogs [filename {scp: scp-url | tftp: tftp-url} | scp: scp-url | tftp: tftp-url]} }
```

Syntax Description	
<b>cores</b>	Applies the action on a core file
<i>filename</i>	Name of the file
<b>scp:</b>	Uses the SCP protocol
<i>scp-url</i>	Enter the SCP URL in the following format: username@A.B.C.D:[/dir]/filename
<b>tftp:</b>	Uses the TFTP protocol

<i>tftp-url</i>	Enter the TFTP URL in the following format: A.B.C.D[/dir]/filename
<b>flash</b>	Applies the action on a flash file
<b>support-bundle</b>	Copies the support bundle to the server
<b>syslogs</b>	Applies the action on the syslog file

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

## delete

To delete a file, use the **delete** command.

**delete** { **/force** | **/recursive** | **/rf** } **cores** *filename*

Syntax Description	
<b>/force</b>	Force delete
<b>/recursive</b>	Recursive delete
<b>/rf</b>	Recursive force delete
<b>cores</b>	Apply action on a core file
<i>filename</i>	Filename to delete

**Command Modes** Privileged EXEC (#)

Command History	Release	Modification
	8.1.111.0	This command was introduced.

### Examples

The following example shows how to delete a file:

```
cisco-ap# delete /rf cores file-name
```

# disable

To turn off privileged commands, use the **disable** command.

## disable

---

### Command Modes

Privileged EXEC (#)

---

### Command History

---

#### Release Modification

8.1.111.0 This command was introduced.

---

### Examples

The following example shows how to turn off privileged commands:

```
cisco-ap# disable
```

# enable

To turn on privileged commands, use the **enable** command.

## enable

---

### Command Modes

User EXEC (>)

---

### Command History

---

#### Release Modification

8.1.111.0 This command was introduced.

---

### Examples

The following example shows how to turn on privileged commands:

```
cisco-ap> enable
```

# exec-timeout

To set the exec-timeout, use the **exec-timeout** command.

**exec-timeout** *timeout-value*

---

### Syntax Description

*timeout-value* Timeout value; valid values range between 0 to 2147483647

---



---

**Command Modes** Privileged EXEC (#)

---

**Command History** **Release** **Modification**

---

8.1.111.0 This command was introduced.

---

### Examples

The following example shows how to set the exec-timeout to 20 seconds:

```
cisco-ap# exec-timeout 20
```

## logging

To log commands, use the **logging** command.

**logging** { **console** [**disable**] | **host** { **clear** | **disable** | **enable** } }

---

**Syntax Description** **console** Console logging

---

**host** Configure syslog server

---

**disable** Disable syslog host logging

---

**enable** Enable syslog server

---

**clear** Clear syslog server IP

---



---

**Command Modes** Privileged EXEC (#)

---

**Command History** **Release** **Modification**

---

8.1.111.0 This command was introduced.

---

### Examples

The following example shows how to enable console logging:

```
cisco-ap# logging console
```

## more

To display a file, use the **more** command.

**more** { **flash** | **syslog** } *file-name*

---

**Syntax Description**    **flash** Apply action on a flash file

---

**syslog** Apply action on syslog file

---

*name* File name

---



---

**Command Modes**    Privileged EXEC (#)

---

**Command History**    **Release**    **Modification**

---

8.1.111.0 This command was introduced.

---

### Examples

The following example shows how to display a syslog file named test-log:

```
cisco-ap# more syslog test-log
```

## reload

To halt the access point or perform a reboot, use the **reload** command.

**reload** [{**at** *hours minutes day-of-month year* | **cancel** | **in** *minutes* | **reason** *reason-string*}]

---

**Syntax Description**    **at**    Reload the AP at a specific date and time

This keyword takes the hour, minute, day of the month, month, and year as parameters; valid values for the keywords are as follows:

- *hour*: 0 to 23
- *minutes*: 0 to 59
- *day-of-the-month*: 1 to 31
- *month*: 1 to 12
- *year*: 2015-2099

---

**cancel** Cancels the pending reload

---

**in**    Reload after a time interval, which you should specify in terms of minutes; valid values are between 1 to 1440 minutes

---

**reason** A string specifying the reason for the reload

---

---

**Command Modes** Privileged EXEC (#)

---

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to reload the AP in 10 minutes:

```
cisco-ap# reload in 10
```

## terminal

To configure terminal parameters, use the **terminal** command.

**terminal** {**length** | **monitor** [**disable** ] | **type** *word* | **width** *no-of-characters*}

---

**Syntax Description**

<b>length</b>	Specifies the number of lines on the screen. Valid values are between 0 to 512. Enter 0 if you do not want the outputs to pause.
<b>monitor</b>	Specifies the debug output to the current terminal line. Press the enter key to enable monitoring. To disable monitoring, enter the keyword <b>disable</b> .
<b>type</b>	Specifies the terminal type
<b>width</b>	Specifies the width of the display terminal; valid values are between 0 to 132

---



---

**Command Modes** Privileged EXEC (#)

---

**Command History**

Release	Modification
8.1.111.0	This command was introduced.

---

### Examples

The following example shows how to configure the terminal length to 50 lines:

```
cisco-ap# terminal length 50
```

