

Troubleshoot COS APs

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Capture Packet Traces \(Sniffer Traces\)](#)

[Wired PCAP on AP Port](#)

[Procedure](#)

[Command Options](#)

[Wired PCAP through the use of Filter](#)

[Radio Capture](#)

[Procedure](#)

[Verify](#)

[Other Options](#)

[Control the AP Client trace from the 9800 WLC](#)

[APs Catalyst 91xx in Sniffer Mode](#)

[Troubleshooting Tips](#)

[Path MTU](#)

[To enable debugs at boot time](#)

[Power save mechanism](#)

[Clients QoS](#)

[Off-Channel scan](#)

[Client Connectivity](#)

[Flexconnect scenarios](#)

[AP Filesystem](#)

[Store and send syslogs](#)

[AP Support Bundle](#)

[Collect AP Core Files Remotely](#)

[AireOS CLI](#)

[AireOS GUI](#)

[Cisco IOS® CLI](#)

[Cisco IOS® GUI](#)

[IoT and Bluetooth](#)

[Conclusion](#)

Introduction

This document describes some of the troubleshooting tools available for Cheatah OS APs (aka COS APs).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document focuses on COS APs like APs models of the series 2800, 3800, 1560 and 4800, as well as new 11ax APs Catalyst 91xx.

This document focuses on many features available in AireOS 8.8 and later. And also Cisco IOS® XE 16.2.2s and later.

There can be comments about availability of certain features in prior releases.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Capture Packet Traces (Sniffer Traces)

Wired PCAP on AP Port

It is possible (as of 8.7 with the filter available in 8.8) to take a pcap on the AP ethernet port. You can either display the result live on the CLI (with only summarized packet details) or save it as a full pcap in the AP flash.

The wired pcap captures everything on the ethernet side (both Rx/Tx) and the tap point inside the AP is immediately before the packet is put on wire.

However, it only captures AP CPU-plane traffic, which means traffic to and from the AP (AP DHCP, AP capwap control tunnel, ...) and does not show client traffic.

Note that the size is very limited (Max size limit of 5MB), so it can be required to configure filters to capture only the traffic you are interested in.

Ensure to stop the traffic capture with "no debug traffic wired ip capture" or simply "undebug all" before you try to copy it (otherwise the copy does not end as packets are still written).

Procedure

Step 1. Start the pcap; select the traffic type with `no debug traffic wired ip capture`:

```
<#root>
```

```
AP70DB.98E1.3DEC#debug traffic wired ip capture
```

```
% Writing packets to "/tmp/pcap/
```

```
AP70DB.98E1.3DEC_capture.pcap0"
```

```
AP70DB.98E1.3DEC#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
```

Step 2. Wait for the traffic to flow and then Stop the capture with the command "no debug traffic wired ip capture" or simply "undebug all":

```
AP70DB.98E1.3DEC#no debug traffic wired ip capture
```

Step 3. Copy the file to tftp/scp server:

```
<#root>
```

```
AP70DB.98E1.3DEC#copy pcap
```

```
AP70DB.98E1.3DEC_capture.pcap0
```

```
tftp 192.168.1.100
```

```
#####
```

```
AP70DB.98E1.3DEC#
```

Step 4. Now you can open the file in wireshark. The file is pcap0. Change to pcap so that it automatically associates with wireshark.

Command Options

The debug traffic wired command has several options that can help you to capture specific traffic:

```
APC4F7.D54C.E77C#debug traffic wired
<0-3>  wired debug interface number
filter filter packets with tcpdump filter string
ip      Enable wired ip traffic dump
tcp     Enable wired tcp traffic dump
udp     Enable wired udp traffic dum
```

You can add "verbose" at the end of the debug command to see the hex dump of the packet. Be aware that this can overwhelm your CLI session very quickly if your filter is not narrow enough.

Wired PCAP through the use of Filter

The filter format corresponds with tcpdump capture filter format.

	Filter Example	Description
Host	host 192.168.2.5	This filters the packet capture to only gather packets which go to or come from the host 192.168.2.5.
	src host 192.168.2.5	This filters the packet capture to only gather packets that come from 192.168.2.5.
	dst host 192.168.2.5	This filters the packet capture to only gather packets which go to 192.168.2.5.

Port	â€œport 443â€	This filters the packet capture to only gather packets with a source or destination of port 443.
	â€œsrc port 1055â€	This captures traffic which is sourced from port 1055.
	â€œdst port 443â€	This captures traffic destined for port 443.

Here is an example where the output displays on the console but also filtered to only see CAPWAP data packets:

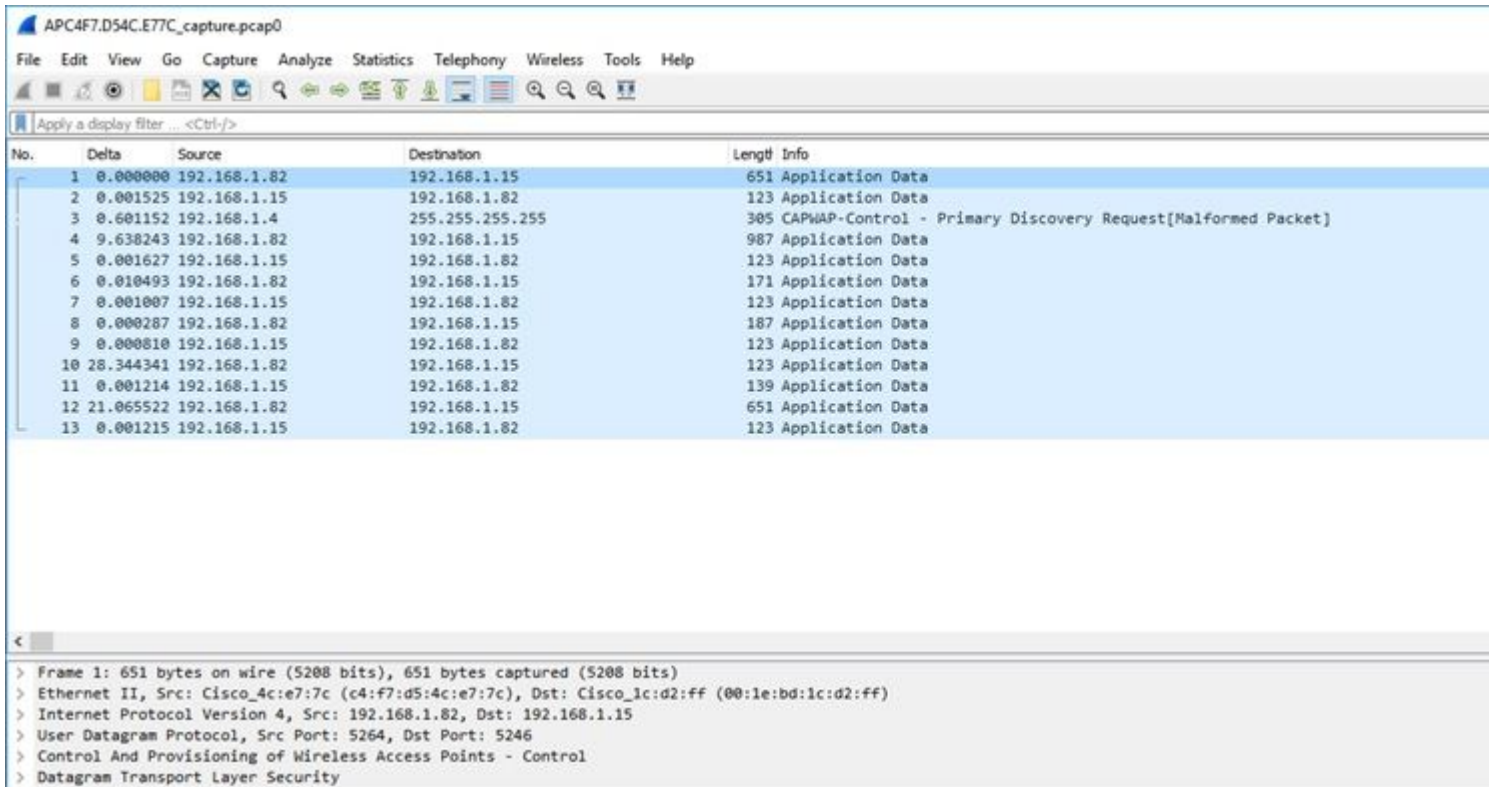
```
APC4F7.D54C.E77C#debug traffic wired filter "port 5246"
APC4F7.D54C.E77C#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
12:20:50.483125 IP APC4F7-D54C-E77C.lan.5264 > 192.168.1.15.5246: UDP, length 81
12:20:50.484361 IP 192.168.1.15.5246 > APC4F7-D54C-E77C.lan.5264: UDP, length 97
```

```
APC4F7.D54C.E77C#no debug traffic wired filter "port 5246"
APC4F7.D54C.E77C#Killed
APC4F7.D54C.E77C#
```

Example of output on File:

```
APC4F7.D54C.E77C#debug traffic wired filter "port 5246" capture
% Writing packets to "/tmp/pcap/APC4F7.D54C.E77C_capture.pcap0"
APC4F7.D54C.E77C#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
APC4F7.D54C.E77C#no debug traffic wired filter "port 5246" capture
APC4F7.D54C.E77C#copy pcap APC4F7.D54C.E77C_capture.pcap0 tftp 192.168.1.100
#####
APC4F7.D54C.E77C#
```

To open the capture on wireshark:



Radio Capture

It is possible to enable the capture of packets on the control-plane of the radio. Due to performance impact, it is not possible to capture on the radio dataplane.

This means that the client association flow (probes, authentication, association, eap, arp, dhcp packets as well as ipv6 control packets, icmp and ndp) is visible but not the data the client passes after the move to the connected state.

Procedure

Step 1. Add the tracked client mac address. Several mac addresses can be added. It is also possible to run the command for all clients but this is not recommended.

```

config ap client-trace address add < client-mac> --- Per client debugging. Allows multiple macs.
config ap client-trace all-clients <enable | disable> -- All clients debugging. Not recommended.

```

Step 2. Set a filter to only log specific protocols or all supported protocols:

```

config ap client-trace filter <all|arp|assoc|auth|dhcp|eap|icmp|ipv6|ndp|probe> <enable|disable>

```

Step 3. Chose to display the output on console (asynchronously):

```

configure ap client-trace output console-log enable

```

Step 4. Start the trace.

```
config ap client-trace start
```

Example:

```
<#root>
```

```
AP0CD0.F894.46E4#show dot11 clients
```

```
Total dot11 clients: 1
```

```
Client MAC Slot ID WLAN ID AID WLAN Name RSSI Maxrate WGB
```

```
A8:DB:03:08:4C:4A
```

```
0 1 1 testewlcwlan -41 MCS92SS No
```

```
AP0CD0.F894.46E4#config ap client-trace address add
```

```
A8:DB:03:08:4C:4A
```

```
AP0CD0.F894.46E4#config ap client-trace filter
```

```
all Trace ALL filters  
arp Trace arp Packets  
assoc Trace assoc Packets  
auth Trace auth Packets  
dhcp Trace dhcp Packets  
eap Trace eap Packets  
icmp Trace icmp Packets  
ipv6 Trace IPv6 Packets  
ndp Trace ndp Packets  
probe Trace probe Packets
```

```
AP0CD0.F894.46E4#config ap client-trace filter all enable
```

```
AP0CD0.F894.46E4#configure ap client-trace output console-log enable
```

```
AP0CD0.F894.46E4#configure ap client-trace start
```

```
AP0CD0.F894.46E4#term mon
```

To stop the capture:

```
configure ap client-trace stop
```

```
configure ap client-trace clear
```

```
configure ap client-trace address clear
```

Verify

Verify Client Trace:

```
<#root>
```

AP70DB.98E1.3DEC#

show ap client-trace status

```
Client Trace Status      : Started
Client Trace ALL Clients : disable
Client Trace Address     : a8:db:03:08:4c:4a
Remote/Dump Client Trace Address : a8:db:03:08:4c:4a

Client Trace Filter      : probe
Client Trace Filter      : auth
Client Trace Filter      : assoc
Client Trace Filter      : eap
Client Trace Filter      : dhcp
Client Trace Filter      : dhcpv6
Client Trace Filter      : icmp
Client Trace Filter      : icmpv6
Client Trace Filter      : ndp
Client Trace Filter      : arp

Client Trace Output      : eventbuf
Client Trace Output      : console-log
Client Trace Output      : dump
Client Trace Output      : remote

Remote trace IP          : 192.168.1.100
Remote trace dest port   : 5688
NOTE - Only VIP packets are seen on remote if VIP is enabled

Dump packet length      : 10
Client Trace Inline Monitor      : disable
Client Trace Inline Monitor pkt-attach : disable
```

Example of a successful client connection:


```

[*04/06/2020 10:11:54.287675] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v1> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:11:54.288144] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:11:54.289870] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ASSOC_REQUEST
[*04/06/2020 10:11:54.317341] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ASSOC_RESPONSE
[*04/06/2020 10:11:54.341370] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] EAPOL_KEY.M1 : Descr
[*04/06/2020 10:11:54.374500] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] EAPOL_KEY.M2 : Descr
[*04/06/2020 10:11:54.377237] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] EAPOL_KEY.M3 : Descr
[*04/06/2020 10:11:54.390255] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] EAPOL_KEY.M4 : Descr
[*04/06/2020 10:11:54.396855] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.416650] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.469089] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.469157] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:57.921877] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:57.921942] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:15:36.123119] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_DEAUTHENTICATI
[*04/06/2020 10:15:36.127731] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_DISASSOC : (.)
[*04/06/2020 10:17:24.128751] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.128870] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v1> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.129303] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.133026] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ASSOC_REQUEST
[*04/06/2020 10:17:24.136095] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ASSOC_RESPONSE
[*04/06/2020 10:17:24.138732] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M1 : Descr
[*04/06/2020 10:17:24.257295] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M2 : Descr
[*04/06/2020 10:17:24.258105] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M3 : Descr
[*04/06/2020 10:17:24.278937] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M4 : Descr
[*04/06/2020 10:17:24.287459] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.301344] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.327482] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.327517] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.430136] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.430202] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:19:08.075326] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_PROBE_REQUEST
[*04/06/2020 10:19:08.075392] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_PROBE_RESPONSE
[*04/06/2020 10:19:08.075437] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v1> [U:W] DOT11_PROBE_REQUEST

```

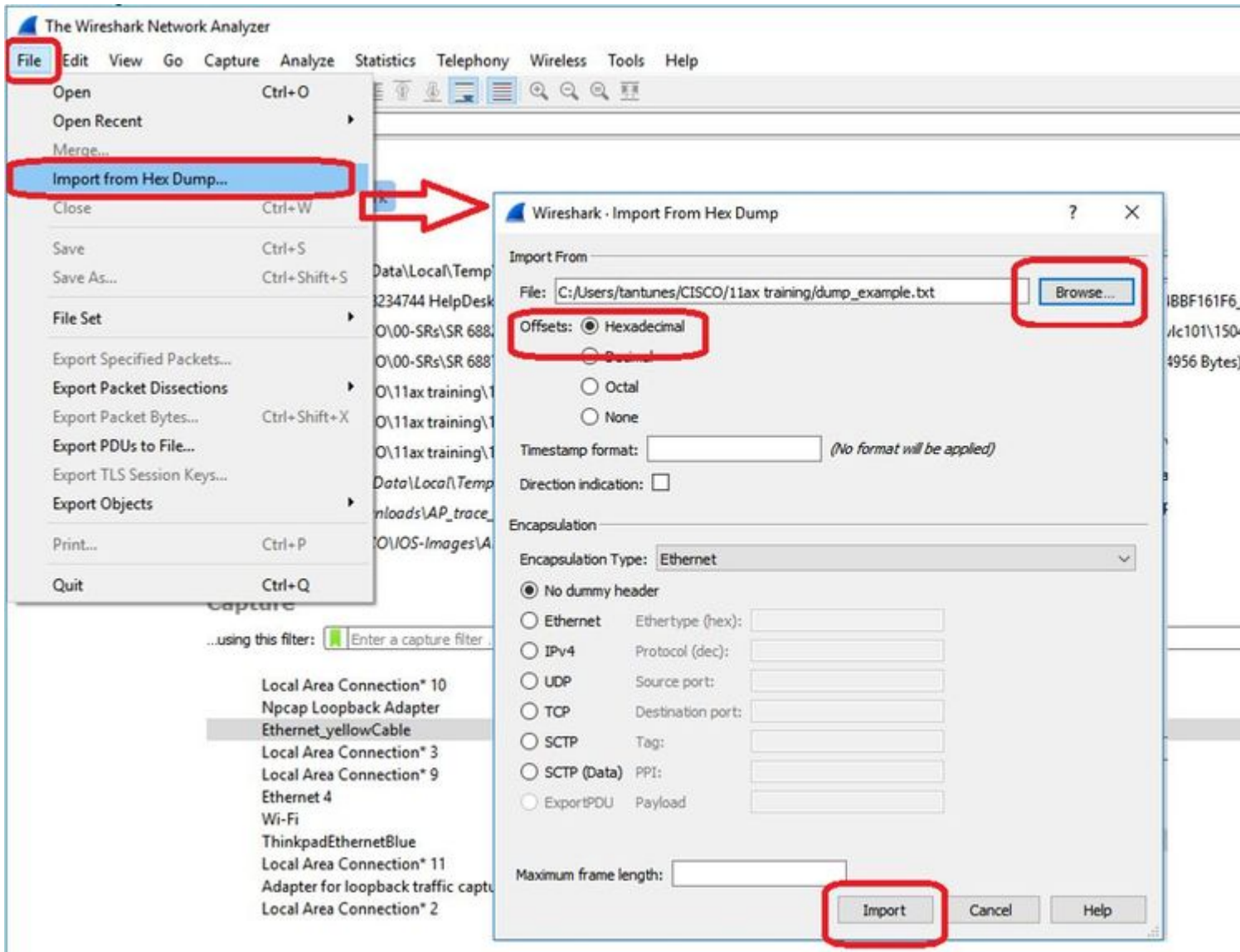
Dump the packets in hex format

You can dump the packets in hex format in the CLI:

```

configure ap client-trace output dump address add xx:xx:xx:xx:xx:xx
configure ap client-trace output dump enable x -> Enter the packet dump length value

```

Because the output can be very large and to consider that the output only mentions what frame type is seen and not any of the inner detail, it can be more efficient to redirect the packet capture to a laptop that run a capture application (such as wireshark).

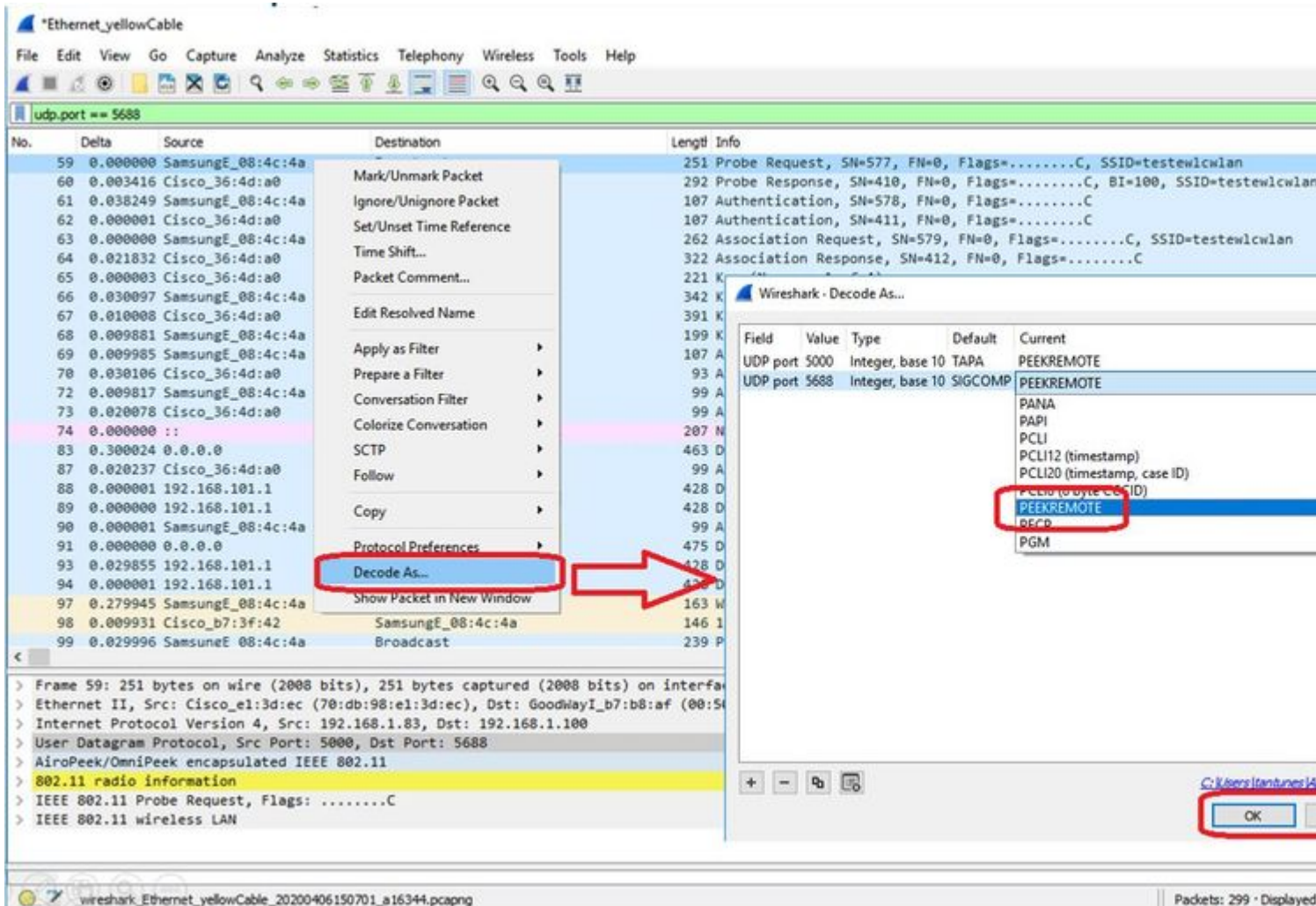
Enable the remote capture feature to send the packets to external device with wireshark:

```
config ap client-trace output remote enable
```

The command means the AP forwards every frame captured by the client-trace filter towards the laptop at 192.168.68.68 and uses PEEKREMOTE encapsulation (just like APs in sniffer mode) on port 5000.

One limitation is that the target laptop has to be in the same subnet as the AP where you run this command on. You can change the port number to accommodate any security policies in place in your network.

Once you received all the packets on the laptop that runs Wireshark, you can right click on the udp 5000 header and chose **decode as** and pick PEEKREMOTE as illustrated in this figure:



List of bugs and enhancements around this feature :

[Cisco bug ID CSCvm09020](#) DNS not seen by client trace anymore on 8.8

[Cisco bug ID CSCvm09015](#) client trace shows many ICMP_other with null sequence number

[Cisco bug ID CSCvm02676](#) AP COS client-trace does not capture webauth packets

Cisco Bug ID [CSCvm02613](#) AP COS client-trace remote output does not work

Cisco Bug ID [CSCvm00855](#) client-trace SEQ numbers inconsistent

Control the AP Client trace from the 9800 WLC

You can configure several APs to do a radio client trace and trigger it from the

Step 1. Configure an AP trace profile that defines which traffic to capture

config term

```
wireless profile ap trace <TRACE-NAME>
  filter all
  no filter probe
  output console-log
```

Step 2. Add the AP trace profile to an AP join profile that is used by the APs that you target.

```
ap profile < ap join profile name>
  trace <TRACE-NAME>
```

Ensure that this ap join profile is applied to a site tag that is used by your target APs

Step 4 Trigger start/stop

```
ap trace client start ap <ap name> client all/<mac>
ap trace client stop ap <ap name> client all/<mac>

ap trace client start site <site tag> client all/<mac>
ap trace client stop site <site tag> client all/<mac>
```

Verification commands :

```
show wireless profile ap trace summary
show wireless profile ap trace detailed PROF_NAME detail
sh ap trace client summary
show ap trace unsupported-ap summary
```

APs Catalyst 91xx in Sniffer Mode

The new Catalyst 9115, 9117, 9120 and 9130 can be configured in sniffer mode. The procedure is similar to previous AP models.

Search Menu Items

- Dashboard
- Monitoring
- Configuration
- Administration
- Troubleshooting

Configuration > Wireless > Access Points

All Access Points

Number of AP(s): 4

AP Name	AP Model	Slots	Admin Status	IP Address
AP70D9.98E1.3DEC	AIR-AP3802I-I-K9	2		192.168.1.83
AP0C00.F894.46E4	C9117AXI-B	2		192.168.1.95
APb4de.318b.fee0	AIR-CAP3702I-I-K9	2		192.168.1.79
APC4F7.D54C.E77C	C9120AXI-B	2		192.168.1.82

- > 5 GHz Radios
- > 2.4 GHz Radios
- > Dual-Band Radios
- > Country
- > LSC Provision

Edit AP

General Interfaces High Availability Inventory

General

AP Name*

Location*

Base Radio MAC

Ethernet MAC

Admin Status

AP Mode

Operation Status

Fabric Status

LED State

LED Brightness Level

CleanAir [NSLKey](#)

Tags

Policy

Site

Search Menu Items

- Dashboard
- Monitoring
- Configuration
- Administration
- Troubleshooting

Configuration > Wireless > Access Points

All Access Points

Number of AP(s): 4

AP Name	AP Model	Slots	Admin Status	IP Address
AP70DB.98E1.3DEC	AIR-AP3802I-I-K9	2	✓	192.168.1.83
AP0CD0.F894.46E4	C9117AXI-B	2	✓	192.168.1.95
APb4de.318b.fee0	AIR-CAP3702I-I-K9	2	✓	192.168.1.79
APC4F7.D54C.E77C	C9120AXI-B	2	✓	192.168.1.82

5 GHz Radios

2.4 GHz Radios

Number of AP(s): 4

AP Name	Slot No	Base Radio MAC	Admin St
AP70DB.98E1.3DEC	0	0027.e336.4da0	✓
AP0CD0.F894.46E4	0	dcd0.f897.03e0	✓
APb4de.318b.fee0	0	b4de.31a4.e030	✓
APC4F7.D54C.E77C	0	c064.e422.1780	✓

Edit Radios 2.4 GHz Band

Configure Detail

Admin Status **ENABLED**

CleanAir Admin Status **ENABLED**

Antenna Parameters

Antenna Type Internal

Antenna A

Antenna B

Antenna C

Antenna D

Antenna Gain 10

Sniffer Channel Assignment

Enable Sniffing

Sniff Channel 6

Sniffer IP* 192.168.1.100

Sniffer IP Status Valid

Download Core Dump to bootflash

Cancel

*ThinkpadEthernetBlue

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 5000

No.	Delta	Source	Destination	Length	Info
2..	0.032866	SamsungE_08:4c:4a	Cisco_97:03:ef	107	Authentication, SN=37, FN=0, Flags=.....C
2..	0.000001	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.001720	Cisco_97:03:ef	SamsungE_08:4c:4a	107	Authentication, SN=0, FN=0, Flags=.....C
2..	0.000301	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.000791	SamsungE_08:4c:4a	Cisco_97:03:ef	360	Association Request, SN=38, FN=0, Flags=.....C, SSI
2..	0.000230	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.004269	Cisco_97:03:ef	SamsungE_08:4c:4a	398	Association Response, SN=1, FN=0, Flags=.....C
2..	0.000750	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.010966	Cisco_97:03:ef	SamsungE_08:4c:4a	221	Key (Message 1 of 4)
2..	0.000001	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.021911	SamsungE_08:4c:4a	Cisco_97:03:ef	342	Key (Message 2 of 4)
2..	0.000002	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.002186	Cisco_97:03:ef	SamsungE_08:4c:4a	391	Key (Message 3 of 4)
2..	0.000935	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]
2..	0.013829	SamsungE_08:4c:4a	Cisco_97:03:ef	199	Key (Message 4 of 4)
2..	0.000174	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]

```

> Tag: Supported Rates 6(8), 9, 12(8), 18, 24(8), 36, 48, 54, [Mbit/sec]
> Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
> Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (44)
> Tag: HT Capabilities (802.11n D1.10)
> Tag: HT Information (802.11n D1.10)
> Tag: Extended Capabilities (8 octets)
> Tag: VHT Capabilities
> Tag: VHT Operation
> Tag: Mobility Domain
> Tag: Fast BSS Transition
> Tag: RM Enabled Capabilities (5 octets)
> Tag: BSS Max Idle Period
< Ext Tag: HE Capabilities (IEEE Std 802.11ax/D3.0)
  Tag Number: Element ID Extension (255)
  Ext Tag length: 46
  Ext Tag Number: HE Capabilities (IEEE Std 802.11ax/D3.0) (35)
  > HE MAC Capabilities Information: 0x800002100009
  > HE Phy Capabilities Information
  < Supported HE-MCS and NSS Set
    < Rx and Tx MCS Maps <= 80 MHz
      < Rx HEX-MCS Map <= 80 MHz: 0xaaaa
        .... ..10 = Max HE-MCS for 1 SS: Support for HE-MCS 0-11 (0x2)
        .... ..10.. = Max HE-MCS for 2 SS: Support for HE-MCS 0-11 (0x2)
        .... ..10... = Max HE-MCS for 3 SS: Support for HE-MCS 0-11 (0x2)
        .... ..10.... = Max HE-MCS for 4 SS: Support for HE-MCS 0-11 (0x2)
        .... ..10..... = Max HE-MCS for 5 SS: Support for HE-MCS 0-11 (0x2)
        .... ..10..... = Max HE-MCS for 6 SS: Support for HE-MCS 0-11 (0x2)
        ..10..... = Max HE-MCS for 7 SS: Support for HE-MCS 0-11 (0x2)
        10..... = Max HE-MCS for 8 SS: Support for HE-MCS 0-11 (0x2)
      > Tx HEX-MCS Map <= 80 MHz: 0xaaaa
    > PPE Thresholds
  < Ext Tag: HE Operation (IEEE Std 802.11ax/D3.0)
    Tag Number: Element ID Extension (255)
    Ext Tag length: 9
    Ext Tag Number: HE Operation (IEEE Std 802.11ax/D3.0) (36)
    > HE Operation Parameters: 0x003ff4
    > BSS Color Information: 0x01
    > Basic HE-MCS and NSS Set: 0xffff

```

Note: Data frames sent at WIFI 6 data rates are captured but, because peekremote is not up to date on Wireshark, they show as 802.11ax phy type as of now. The fix is in Wireshark 3.2.4 where Wireshark displays the proper wifi6 phy rate.

Note: Cisco APs can't capture MU-OFDMA frames at this time but can capture the trigger frames (sent at management data rate) that announce a MU-OFDMA window. You can already infer that MU-OFDMA happens (or not) and with which client.

Troubleshooting Tips

Path MTU

Although Path MTU discovery finds the optimal MTU for the AP, it is possible to override this settings manually.

On AireOS 8.10.130 WLC, the command **config ap pmtu disable <ap/all>** sets a static MTU for one or all APs rather than to rely on the dynamic discovery mechanism.

To enable debugs at boot time

You can run `config boot debug capwap` to enable capwap,DTLS and DHCP debugs at the next boot time, even before the OS has booted and the prompt is shown.

You also have "config boot debug memory xxxx" for several memory debugs.

You can see if boot debugs are enabled or not at next reboot with "show boot".

They can be disabled with the addition of the disable keyword at the end such as "config boot debug capwap disable".

Power save mechanism

The power save of a given client can be troubleshot by running

debug client trace <mac address>

Clients QoS

To verify that QoS tags are applied, you can run "*debug capwap client qos*".

It displays the UP value of packets for wireless clients.

It is not mac filterable as of 8.8 (enhancement request Cisco bug ID [CSCvm08899](#)).

```
labAP#debug capwap client qos
```

```
[*08/20/2018 09:43:36.3171] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:89
[*08/20/2018 09:43:45.0051] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:89
[*08/20/2018 09:43:45.5463] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:89
[*08/20/2018 09:43:46.5687] chatter: set_qos_up :: SetQosPriority: bridged packet dst: AC:81:12:C7:CD:35
[*08/20/2018 09:43:47.0982] chatter: set_qos_up :: SetQosPriority: bridged packet dst: AC:81:12:C7:CD:35
```

You can also verify the Qos UP to DSCP table on the AP as well as total amount of packets marked, shaped and dropped by Qos:

```
LabAP#show dot11 qos
Qos Policy Maps (UPSTREAM)
```

```
no policymap
Qos Stats (UPSTREAM)
```

```
total packets: 0
dropped packets: 0
marked packets: 0
shaped packets: 0
policed packets: 0
copied packets: 0
```

```
DSCP TO DOT1P (UPSTREAM)
```

```
Default dscp2dot1p Table Value:
[0]->0 [1]->2 [2]->10 [3]->18 [4]->26 [5]->34 [6]->46 [7]->48
Active dscp2dot1p Table Value:
[0]->0 [1]->2 [2]->10 [3]->18 [4]->26 [5]->34 [6]->46 [7]->48
```

```
Qos Policy Maps (DOWNSTREAM)
```

```
no policymap
Qos Stats (DOWNSTREAM)
```

```
total packets: 0
dropped packets: 0
marked packets: 0
shaped packets: 0
policed packets: 0
copied packets: 0
```

```
DSCP TO DOT1P (DOWNSTREAM)
```

```
Default dscp2dot1p Table Value:
[0]->0 [1]->-1 [2]->1 [3]->-1 [4]->1 [5]->-1 [6]->1 [7]->-1
[8]->-1 [9]->-1 [10]->2 [11]->-1 [12]->2 [13]->-1 [14]->2 [15]->-1
[16]->-1 [17]->-1 [18]->3 [19]->-1 [20]->3 [21]->-1 [22]->3 [23]->-1
[24]->-1 [25]->-1 [26]->4 [27]->-1 [28]->-1 [29]->-1 [30]->-1 [31]->-1
[32]->-1 [33]->-1 [34]->5 [35]->-1 [36]->-1 [37]->-1 [38]->-1 [39]->-1
[40]->-1 [41]->-1 [42]->-1 [43]->-1 [44]->-1 [45]->-1 [46]->6 [47]->-1
[48]->7 [49]->-1 [50]->-1 [51]->-1 [52]->-1 [53]->-1 [54]->-1 [55]->-1
[56]->7 [57]->-1 [58]->-1 [59]->-1 [60]->-1 [61]->-1 [62]->-1 [63]->-1
Active dscp2dot1p Table Value:
[0]->0 [1]->-1 [2]->1 [3]->-1 [4]->1 [5]->-1 [6]->1 [7]->-1
[8]->-1 [9]->-1 [10]->2 [11]->-1 [12]->2 [13]->-1 [14]->2 [15]->-1
[16]->-1 [17]->-1 [18]->3 [19]->-1 [20]->3 [21]->-1 [22]->3 [23]->-1
[24]->-1 [25]->-1 [26]->4 [27]->-1 [28]->-1 [29]->-1 [30]->-1 [31]->-1
[32]->-1 [33]->-1 [34]->5 [35]->-1 [36]->-1 [37]->-1 [38]->-1 [39]->-1
[40]->-1 [41]->-1 [42]->-1 [43]->-1 [44]->-1 [45]->-1 [46]->6 [47]->-1
[48]->7 [49]->-1 [50]->-1 [51]->-1 [52]->-1 [53]->-1 [54]->-1 [55]->-1
[56]->7 [57]->-1 [58]->-1 [59]->-1 [60]->-1 [61]->-1 [62]->-1 [63]->-1
LabAP#
```

When Qos policies are defined on the WLC and downloaded on the Flexconnect AP, you can verify them with :

```
AP780C-F085-49E6#show policy-map
2 policymaps
Policy Map BWLimitAAAClients          type:qos client:default
  Class BWLimitAAAClients_AVC_UI_CLASS
```

drop

```
Class BWLimitAAAClients_ADV_UI_CLASS
  set dscp af41 (34)
```

```
Class class-default
  police rate 5000000 bps (625000Bytes/s)
  conform-action
  exceed-action
```

```
Policy Map platinum-up          type:qos client:default
  Class cm-dscp-set1-for-up-4
    set dscp af41 (34)
```

```
Class cm-dscp-set2-for-up-4
  set dscp af41 (34)
```

```
Class cm-dscp-for-up-5
  set dscp af41 (34)
```

```
Class cm-dscp-for-up-6
  set dscp ef (46)
```

```
Class cm-dscp-for-up-7
  set dscp ef (46)
```

```
Class class-default
  no actions
```

In case of Qos rate-limiting :

```
AP780C-F085-49E6#show rate-limit client
```

Config:

```
          mac vap rt_rate_out rt_rate_in rt_burst_out rt_burst_in nrt_rate_out nrt_rate_in nrt_burst
A8:DB:03:6F:7A:46 2          0          0          0          0          0          0          0
```

Statistics:

	name	up	down
	Unshaped	0	0
	Client RT pass	0	0
	Client NRT pass	0	0
	Client RT drops	0	0
	Client NRT drops	0	38621
		9 54922	0

Off-Channel scan

Debugging the off-channel scan of the AP can be useful when troubleshooting rogue detection (to validate if

and when the AP goes on a specific channel to scan), but can also be useful in video troubleshoot where a sensitive real-time stream gets constant interruptions if the "off channel scan defer" feature is not used.

```
debug rrm off-channel defer
debug rrm off-channel dbg (starting 17.8.1)
debug rrm off-channel schedule
debug rrm off-channel voice (starting 17.8.1)
debug rrm schedule (starting 17.8.1, debug NDP packet tx)
show trace dot_11 channel enable
```

```
[*06/11/2020 09:45:38.9530] wcp/rrm_userspace_0/rrm_schedule :: RRMSchedule process_int_duration_timer_1
[*06/11/2020 09:45:39.0550] noise measurement channel 5 noise 89
[*06/11/2020 09:45:43.5490] wcp/rrm_userspace_1/rrm_schedule :: RRMSchedule process_int_duration_timer_1
[*06/11/2020 09:45:43.6570] noise measurement channel 140 noise 97
```

Client Connectivity

It is possible to list clients that have been deauthenticated by the access point with the last event timestamp:

```
LabAP#show dot11 clients deauth
      timestamp          mac vap reason_code
Mon Aug 20 09:50:59 2018 AC:BC:32:A4:2C:D3 9      4
Mon Aug 20 09:52:14 2018 00:AE:FA:78:36:89 9      4
Mon Aug 20 10:31:54 2018 00:AE:FA:78:36:89 9      4
```

In the previous output, the reason code is the deauthentication reason code as detailed in this link :

<https://community.cisco.com:443/t5/wireless-mobility-knowledge-base/802-11-association-status-802-11-deauth-reason-codes/ta-p/3148055>

The vap refers to the identifier of the WLAN inside the AP (which is different from the WLAN ID on the WLC !!!).

You can cross-relate it with other outputs detailed subsequently which always mentions the vap of associated clients.

You can see the list of VAP ids with "*show controllers Dot11Radio 0/1 wlan*".

When clients are still associated, you can get details on their connection with:

```
LabAP#show dot11 clients

Total dot11 clients: 1
      Client MAC Slot ID WLAN ID AID WLAN Name RSSI Maxrate WGB
00:AE:FA:78:36:89      1      10  1  TestSSID -25 MCS82SS No
```

A lot more details can be obtained about the client entry with:

LabAP#show client summ

Radio Driver client Summary:

=====

wifi0

[*08/20/2018 11:54:59.5340]
[*08/20/2018 11:54:59.5340] Total STA List Count 0
[*08/20/2018 11:54:59.5340] | NO| MAC|STATE|
[*08/20/2018 11:54:59.5340] -----

wifi1

[*08/20/2018 11:54:59.5357]
[*08/20/2018 11:54:59.5357] Total STA List Count 1
[*08/20/2018 11:54:59.5357] | NO| MAC|STATE|
[*08/20/2018 11:54:59.5357] -----
[*08/20/2018 11:54:59.5357] | 1| 0:ffffffae:fffffffa:78:36:ffffff89| 8|

Radio Driver Client AID List:

=====

wifi0

[*08/20/2018 11:54:59.5415]
[*08/20/2018 11:54:59.5415] Total STA-ID List Count 0
[*08/20/2018 11:54:59.5415] | NO| MAC|STA-ID|
[*08/20/2018 11:54:59.5415] -----

wifi1

[*08/20/2018 11:54:59.5431]
[*08/20/2018 11:54:59.5431] Total STA-ID List Count 1
[*08/20/2018 11:54:59.5431] | NO| MAC|STA-ID|
[*08/20/2018 11:54:59.5432] -----
[*08/20/2018 11:54:59.5432] | 1| 0:ffffffae:fffffffa:78:36:ffffff89| 6|

WCP client Summary:

=====

mac radio vap aid state encr Maxrate is_wgb_wired wgb_mac_addr
00:AE:FA:78:36:89 1 9 1 FWD AES_CCM128 MCS82SS false 00:00:00:00:00:00

NSS client Summary:

=====

Current Count: 3

MAC	OPAQUE	PRI POL	VLAN	BR	TN	QCF	BSS	RADID	MYMAC
F8:0B:CB:E4:7F:41	00000000	3	0	1	1	0	2	3	1
F8:0B:CB:E4:7F:40	00000000	3	0	1	1	0	2	3	1
00:AE:FA:78:36:89	00000003	1	0	1	1	0	9	1	0

Datapath IPv4 client Summary:

=====

id vap port node tunnel mac seen_ip hashed_ip sniff_ac
00:AE:FA:78:36:89 9 apr1v9 192.0.2.13 - 00:AE:FA:78:36:89 192.168.68.209 10.228.153.45 5.990000

Datapath IPv6 client Summary:

=====

client mac seen_ip6 age scope port
1 00:AE:FA:78:36:89 fe80::2ae:faff:fe78:3689 61 link-local apr1v9

Wired client Summary:

=====

mac port state local_client detect_ago associated_ago tx_pkts tx_bytes rx_pkts rx_bytes

You can force the disconnection of a specific client with :

test dot11 client deauthenticate

Traffic counters can be obtained per-client with:

```
LabAP#show client statistics wireless 00:AE:FA:78:36:89
Client MAC address: 00:AE:FA:78:36:89
Tx Packets           : 621
Tx Management Packets : 6
Tx Control Packets   : 153
Tx Data Packets      : 462
Tx Data Bytes        : 145899
Tx Unicast Data Packets : 600
Rx Packets           : 2910
Rx Management Packets : 13
Rx Control Packets   : 943
Rx Data Packets      : 1954
Rx Data Bytes        : 145699
LabAP#
```

More on the radio level, a lot of information can be obtained in the "*show controllers*". When you add the client mac address, the supported data rates, current data rates, PHY capabilities as well as amount of retries and txfails, are displayed:

<#root>

```
LabAP#show controllers dot11Radio 0 client 00:AE:FA:78:36:89
      mac radio vap aid state      encr Maxrate is_wgb_wired      wgb_mac_addr
00:AE:FA:78:36:89  0  9  1  FWD AES_CCM128  M15          false 00:00:00:00:00:00
Configured rates for client 00:AE:FA:78:36:89
Legacy Rates(Mbps): 11
HT Rates(MCS):M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15
VHT Rates: 1SS:M0-7 2SS:M0-7

HT:yes      VHT:yes      HE:no      40MHz:no      80MHz:no      80+80MHz:no      160MHz:no
11w:no      MFP:no      11h:no      encrypt_polocy: 4
_wmm_enabled:yes      qos_capable:yes      WME(11e):no      WMM_MIXED_MODE:no
short_preamble:yes      short_slot_time:no      short_hdr:yes      SM_dyn:yes
short_GI_20M:yes      short_GI_40M:no      short_GI_80M:yes      LDPC:yes      AMSDU:yes      AMSDU_long:no
su_mimo_capable:yes      mu_mimo_capable:no      is_wgb_wired:no      is_wgb:no

Additional info for client 00:AE:FA:78:36:89
RSSI: -90
PS : Legacy (Sleeping)
Tx Rate: 0 Kbps
Rx Rate: 117000 Kbps
VHT_TXMAP: 0
CCX Ver: 4

Statistics for client 00:AE:FA:78:36:89
      mac      intf TxData TxMgmt TxUC TxBytes

TxFail

TxDcrd TxCumRetries RxData RxMgmt RxBytes RxErr TxRt  RxRt idle_counter stats_ago expiration
```

00:AE:FA:78:36:89 apr0v9 8 1 6 1038 1 0 0 31 1 1599

Per TID packet statistics for client 00:AE:FA:78:36:89

Priority	Rx Pkts	Tx Pkts	Rx(last 5 s)	Tx (last 5 s)	QID	Tx Drops	Tx Cur	Qlimit
0	899	460	1	1	144	0	0	1024
1	0	0	0	0	145	0	0	1024
2	0	0	0	0	146	0	0	1024
3	59	0	0	0	147	0	0	1024
4	0	0	0	0	148	0	0	1024
5	0	0	0	0	149	0	0	1024
6	0	0	0	0	150	0	0	1024
7	0	0	0	0	151	0	0	1024

Legacy Rate Statistics:

(Mbps : Rx, Tx, Tx-Retries)
11 Mbps : 2, 0, 0
6 Mbps : 0, 9, 0

HT/VHT Rate Statistics:

(Rate/SS/Width : Rx, Rx-Ampdu, Tx, Tx-Ampdu, Tx-Retries)
0/1/20 : 4, 4, 0, 0, 0
6/2/20 : 4, 4, 0, 0, 0
7/2/20 : 5, 5, 0, 0, 0

webauth done:
false

In order to constantly keep track of a client data rate and/or RSSI value, you can run "**debug dot11 client rate address <mac>**" and this logs this information every second:

LabAP#debug dot11 client rate address 00:AE:FA:78:36:89

[*08/20/2018 14:17:28.0928]	MAC	Tx-Pkts	Rx-Pkts	Tx-Rate	Rx-Rate	RSSI	SNR	Tx-R
[*08/20/2018 14:17:28.0928]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-45	53	
[*08/20/2018 14:17:29.0931]	00:AE:FA:78:36:89	7	18	12	a8.2-2s	-45	53	
[*08/20/2018 14:17:30.0934]	00:AE:FA:78:36:89	3	18	12	a8.2-2s	-45	53	
[*08/20/2018 14:17:31.0937]	00:AE:FA:78:36:89	2	20	12	a8.2-2s	-45	53	
[*08/20/2018 14:17:32.0939]	00:AE:FA:78:36:89	2	20	12	a8.2-2s	-45	53	
[*08/20/2018 14:17:33.0942]	00:AE:FA:78:36:89	2	21	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:34.0988]	00:AE:FA:78:36:89	1	4	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:35.0990]	00:AE:FA:78:36:89	9	23	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:36.0993]	00:AE:FA:78:36:89	3	7	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:37.0996]	00:AE:FA:78:36:89	2	6	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:38.0999]	00:AE:FA:78:36:89	2	14	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:39.1002]	00:AE:FA:78:36:89	2	10	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:40.1004]	00:AE:FA:78:36:89	1	6	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:41.1007]	00:AE:FA:78:36:89	9	20	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:42.1010]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:43.1013]	00:AE:FA:78:36:89	2	8	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:44.1015]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:45.1018]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:46.1021]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:47.1024]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:48.1026]	00:AE:FA:78:36:89	7	15	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:49.1029]	00:AE:FA:78:36:89	0	6	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:50.1032]	00:AE:FA:78:36:89	0	0	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:51.1035]	00:AE:FA:78:36:89	1	7	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:52.1037]	00:AE:FA:78:36:89	0	17	12	a8.2-2s	-46	52	
[*08/20/2018 14:17:53.1040]	00:AE:FA:78:36:89	1	19	12	a8.2-2s	-46	52	

[*08/20/2018 14:17:54.1043]	00:AE:FA:78:36:89	2	17	12	a8.2-2s	-46	52
[*08/20/2018 14:17:55.1046]	00:AE:FA:78:36:89	2	22	12	a8.2-2s	-45	53
[*08/20/2018 14:17:56.1048]	00:AE:FA:78:36:89	1	18	12	a8.2-2s	-45	53
[*08/20/2018 14:17:57.1053]	00:AE:FA:78:36:89	2	18	12	a8.2-2s	-45	53
[*08/20/2018 14:17:58.1055]	00:AE:FA:78:36:89	12	37	12	a8.2-2s	-45	53

In this output, the Tx and Rx packet counters are packets transmitted in the second interval since it last printed, same thing for the Tx Retries. However the RSSI, SNR and data rate are the values from the last packet of that interval (and not an average for all packets in that interval).

Flexconnect scenarios

You can verify what ACLs are currently applied to a client in a pre-auth (CWA for example) or post-auth scenario:

```
AP#show client access-lists pre-auth all f48c.507a.b9ad
Pre-Auth URL ACLs for Client: F4:8C:50:7A:B9:AD
IPv4 ACL: IPv6 ACL:
ACTION URL-LIST
```

```
Resolved IPs for Client: F4:8C:50:7A:B9:AD
HIT-COUNT URL ACTION IP-LIST
```

```
REDIRECT
rule 0: allow true and ip proto 17 and src port 53
rule 1: allow true and ip proto 17 and dst port 53
rule 2: allow true and src 10.48.39.161mask 255.255.255.255
rule 3: allow true and dst 10.48.39.161mask 255.255.255.255
rule 4: deny true
No IPv6 ACL found
```

```
AP#show client access-lists post-auth all f48c.507a.b9ad
Post-Auth URL ACLs for Client: F4:8C:50:7A:B9:AD
IPv4 ACL: IPv6 ACL:
ACTION URL-LIST
```

```
Resolved IPs for Client: F4:8C:50:7A:B9:AD
HIT-COUNT URL ACTION IP-LIST
```

```
post-auth
rule 0: deny true and dst 192.0.0.0mask 255.0.0.0
rule 1: deny true and src 192.0.0.0mask 255.0.0.0
rule 2: allow true
No IPv6 ACL found
```

AP Filesystem

COS APs do not allow to list all the content of the file system as on unix platforms.

The command "*show filesystems*" gives a detail of the space usage and distribution on the current partition:


```
2802#show filesystems
Filesystem      Size      Used Available Use% Mounted on
/dev/ubivol/storage 57.5M    364.0K    54.1M    1% /storage
2802#
```

The command "*show flash*" lists the main files on the AP flash. You can also append the syslog or core keyword to list those specific folders.

```
ap_2802#show flash
Directory of /storage/
total 84
-rw-r--r--    1 root    root           0 May 21  2018 1111
-rw-r--r--    1 root    root           6 Apr 15 11:09 BOOT_COUNT
-rw-r--r--    1 root    root           6 Apr 15 11:09 BOOT_COUNT.reserve
-rw-r--r--    1 root    root          29 Apr 15 11:09 RELOADED_AT_UTC
drwxr-xr-x    2 root    root          160 Mar 27 13:53 ap-images
drwxr-xr-x    4 5      root         2016 Apr 15 11:10 application
-rw-r--r--    1 root    root         6383 Apr 26 09:32 base_capwap_cfg_info
-rw-r--r--    1 root    root           20 Apr 26 10:31 bigacl
-rw-r--r--    1 root    root         1230 Mar 27 13:53 bootloader.log
-rw-r--r--    1 root    root           5 Apr 26 09:29 bootloader_verify.shadow
-rw-r--r--    1 root    root           18 Jun 30  2017 config
-rw-r--r--    1 root    root         8116 Apr 26 09:32 config.flex
-rw-r--r--    1 root    root           21 Apr 26 09:32 config.flex.mgroup
-rw-r--r--    1 root    root           0 Apr 15 11:09 config.local
-rw-r--r--    1 root    root           0 Jul 26  2018 config.mesh.dhcp
-rw-r--r--    1 root    root          180 Apr 15 11:10 config.mobexp
-rw-r--r--    1 root    root           0 Jun 5  2018 config.oep
-rw-r--r--    1 root    root         2253 Apr 26 09:43 config.wireless
drwxr-xr-x    2 root    root          160 Jun 30  2017 cores
drwxr-xr-x    2 root    root          320 Jun 30  2017 dropbear
drwxr-xr-x    2 root    root          160 Jun 30  2017 images
-rw-r--r--    1 root    root          222 Jan  2  2000 last_good_uplink_config
drwxr-xr-x    2 root    root          160 Jun 30  2017 lists
-rw-r--r--    1 root    root          215 Apr 16 11:01 part1_info.ver
-rw-r--r--    1 root    root          215 Apr 26 09:29 part2_info.ver
-rw-r--r--    1 root    root         4096 Apr 26 09:36 random_seed
-rw-r--r--    1 root    root           3 Jun 30  2017 rxtx_mode
-rw-r--r--    1 root    root           64 Apr 15 11:11 sensord_CSPRNG0
-rw-r--r--    1 root    root           64 Apr 15 11:11 sensord_CSPRNG1
drwxr-xr-x    3 support  root          224 Jun 30  2017 support
drwxr-xr-x    2 root    root          2176 Apr 15 11:10 syslogs
-----
Filesystem      Size      Used Available Use% Mounted on
flash           57.5M    372.0K    54.1M    1% /storage
```

Store and send syslogs

The syslog folder stores the syslog output from previous reboots. The command "*show log*" only shows syslog since the last reboot.

At each reboot cycle, the syslogs are written on incremental files.

```

artaki# show flash syslogs
Directory of /storage/syslogs/
total 128
-rw-r--r--  1 root    root      11963 Jul  6 15:23 1
-rw-r--r--  1 root    root     20406 Jan  1  2000 1.0
-rw-r--r--  1 root    root       313 Jul  6 15:23 1.last_write
-rw-r--r--  1 root    root     20364 Jan  1  2000 1.start
-rw-r--r--  1 root    root       33 Jul  6 15:23 1.watchdog_status
-rw-r--r--  1 root    root     19788 Jul  6 16:46 2
-rw-r--r--  1 root    root     20481 Jul  6 15:23 2.0
-rw-r--r--  1 root    root       313 Jul  6 16:46 2.last_write
-rw-r--r--  1 root    root     20422 Jul  6 15:23 2.start

```

```

-----
Filesystem      Size      Used Available Use% Mounted on
flash           57.6M    88.0K     54.5M    0% /storage

```

```

artaki# show flash cores
Directory of /storage/cores/
total 0

```

```

-----
Filesystem      Size      Used Available Use% Mounted on
flash           57.6M    88.0K     54.5M    0% /storage

```

The first output after initial boot is file 1.0 and a file 1.1 is created if 1.0 becomes too long. After reboot, a new file 2.0 is created and so on.

From the WLC, you can configure the Syslog destination if you want your APs to send their syslog messages unicast to a specific server.

By default, APs send their syslogs to a broadcast address which can cause quite some broadcast storm, so ensure to configure a syslog server.

The AP sends via syslog by default whatever prints on its console output.

On 9800 Controller, you can change these parameters in the Configuration -> AP Join profile, under Management.

Edit AP Join Profile

General Client CAPWAP AP **Management** Security ICap QoS

Device User Credentials CDP Interface

TFTP Downgrade

IPv4/IPv6 Address

Image File Name

System Log

Facility Value

Host IPv4/IPv6 Address

Log Trap Value

Secured ⓘ

Telnet/SSH Configuration

Telnet

SSH

AP Core Dump

Enable Core Dump

You can change the **Log Trap Value** to also send debugs via syslog. You can then enable debugs on the AP CLI and the output of these are sent via syslog messages to your configured server .

Due to Cisco Bug ID [CSCvu75017](#) ,only when you set the syslog facility to KERN (the default value) does the AP send syslog messages out.

If you are troubleshooting issues where an AP possibly loses network connectivity (or on a WGB for example), syslog is not as reliable as no messages are sent if the AP loses its uplink connectivity.

Therefore, reliance on the stored syslog files in flash is a great way to debug and store the output on the AP itself and then periodically upload it later on.

AP Support Bundle

Some commonly collected diagnostic information of various types can be made available in a single bundle that you can upload from Access Points.

The diagnostic information that can you can include in the bundle are:

- AP show tech
- AP syslogs
- AP Capwapd Brain logs
- AP Startup & Message logs

- AP Coredump files

To get the AP support bundle you can go into the AP CLI and enter the command "**copy support-bundle tftp: x.x.x.x**".

After this you can check for the file named with AP name appended with the **support.apversion.date.time.tgz** as shown subsequently :

```
APC4F7.D54C.E77C#copy support-bundle tftp: 192.168.1.100
<cr>
APC4F7.D54C.E77C#copy support-bundle tftp: 192.168.1.100
Creating support bundle, please wait...ifconfig: wired1: error fetching interface information: Device not found
Unit systemd-journald.socket could not be found.
tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tgz created ===+
=====
Successful file transfer:
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tgz
APC4F7.D54C.E77C#
```

When you "untar" the file you can view the various files collected:

i-Images > APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526

Name	Date modified	Type	Size
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.brain.error.log.gz	4/8/2020 4:55 PM	GZ File	1 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.brain.log.gz	4/8/2020 4:55 PM	GZ File	3 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.info	4/8/2020 4:55 PM	INFO File	1 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.messages.gz	4/8/2020 4:55 PM	GZ File	11 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.startlog.gz	4/8/2020 4:55 PM	GZ File	5 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.syslogs.gz	4/8/2020 4:55 PM	GZ File	2 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tech_support.gz	4/8/2020 4:55 PM	GZ File	34 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.wsa_info.json.gz	4/8/2020 4:55 PM	GZ File	1 KB
<input type="checkbox"/> APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.wsa_status.json.gz	4/8/2020 4:55 PM	GZ File	1 KB

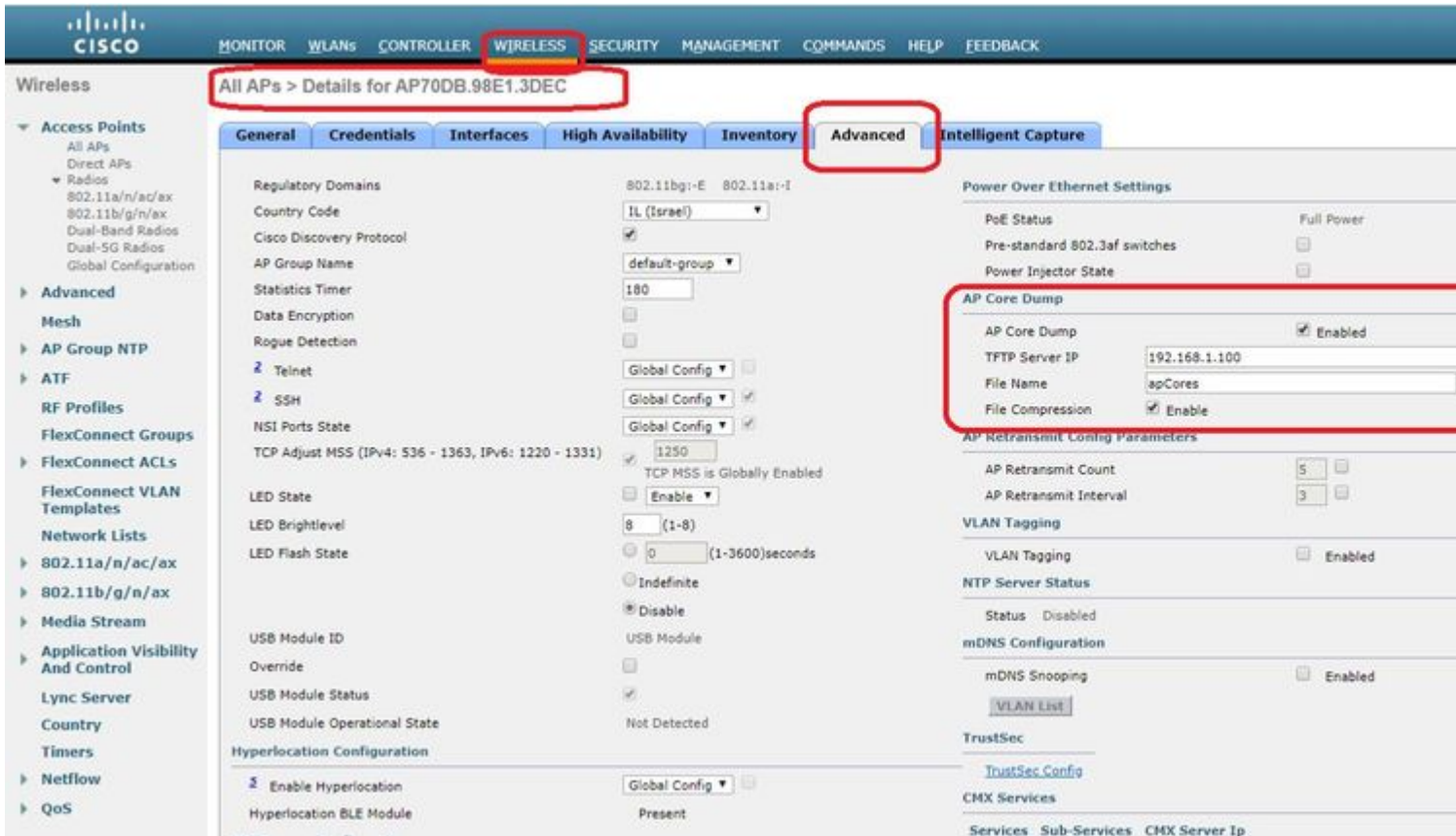
Collect AP Core Files Remotely

To collect AP core files remotely, please enable core dump to be included in support bundle and then Upload support bundle from the AP, or send directly to tftp server. The subsequent examples use tftp server 192.168.1.100.

AireOS CLI

```
(c3504-01) >config ap core-dump enable 192.168.1.100 apCores uncompress ?
<Cisco AP> Enter the name of the Cisco AP.
all Applies the configuration to all connected APs.
```

AireOS GUI



Cisco IOS® CLI

```
<#root>
```

```
eWLC-9800-01(
```

```
config
```

```
)#ap profile TiagoOffice
```

```
eWLC-9800-01(
```

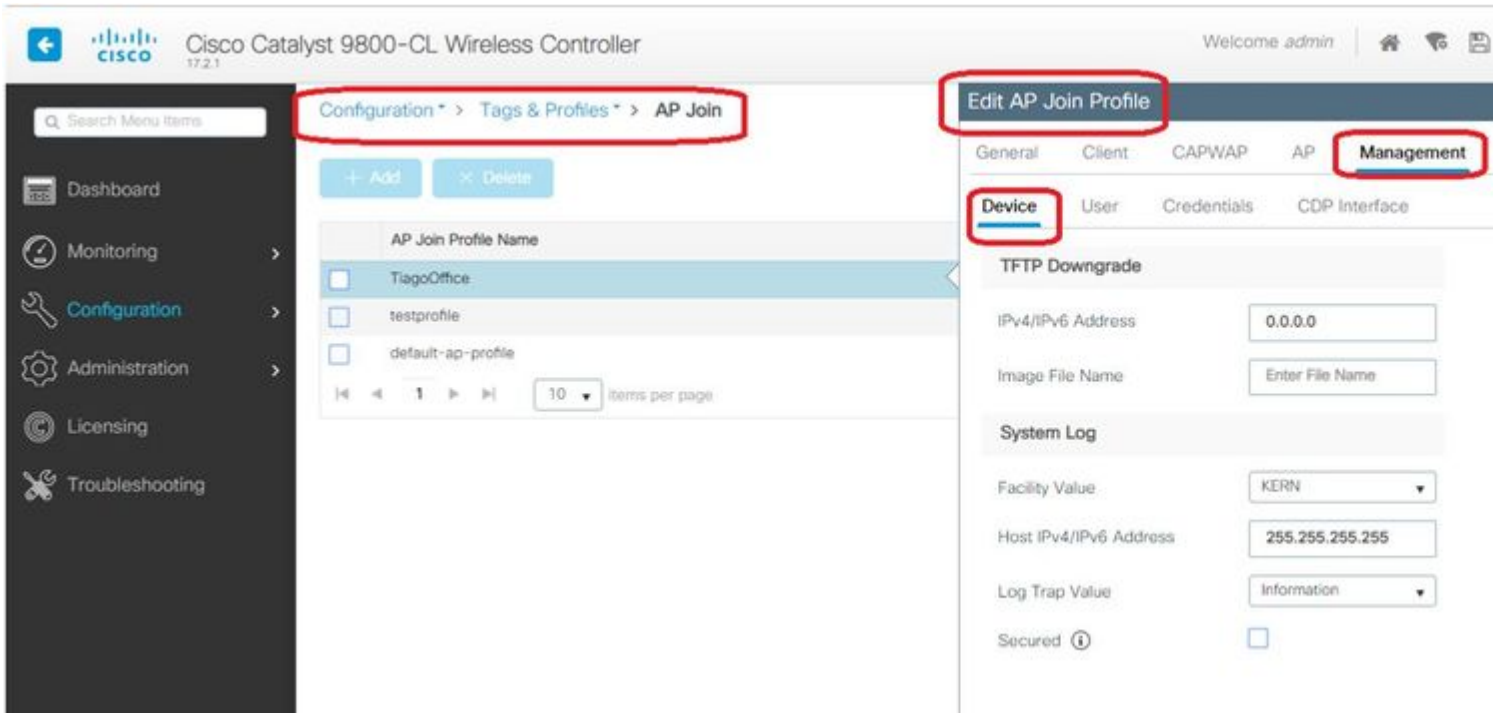
```
config-
```

```
ap
```

```
-profile
```

```
)#core-dump tftp-server 192.168.1.100 file apCores uncompress
```

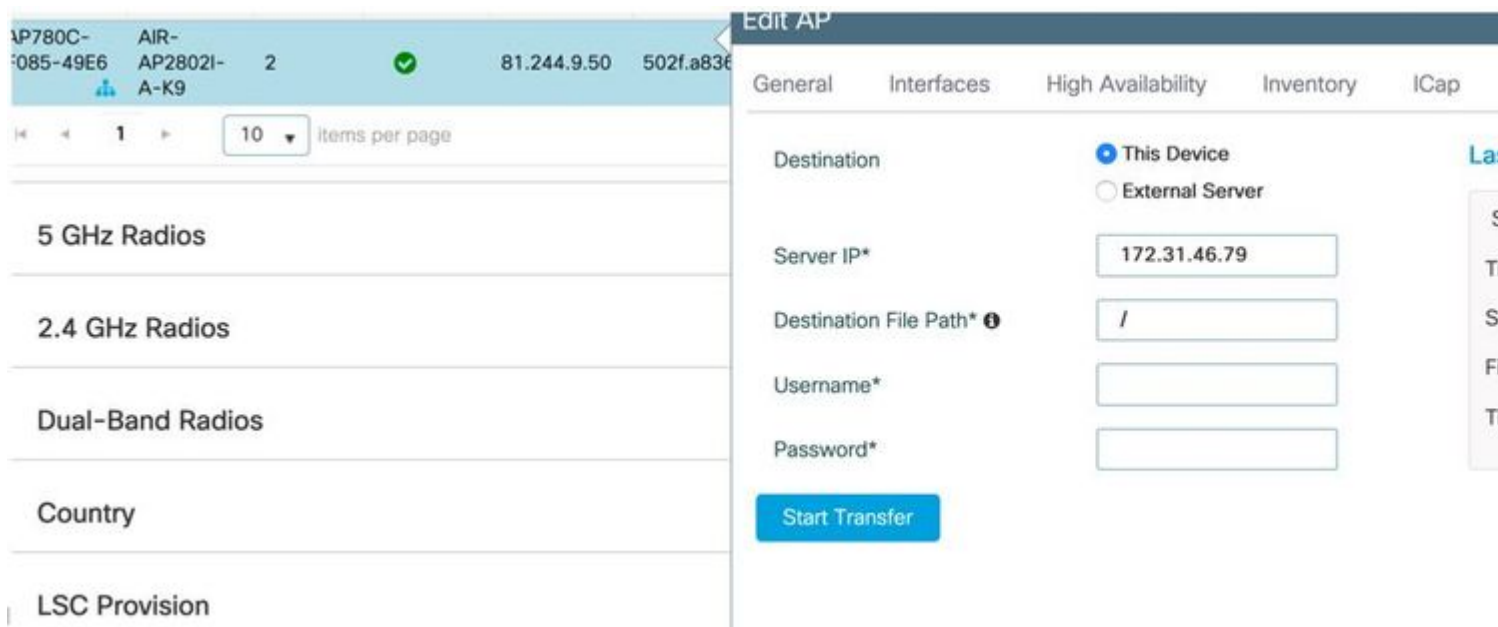
Cisco IOS® GUI



As from Cisco IOS® XE 17.3.1, you have a Support Bundle tab and can download the AP SB from the WLC GUI.

All it does is execute `copy support-bundle` command on the AP and sends it via SCP to the WLC (because WLC can be an SCP server).

And then you can download it from your browser:



This means you can manually do the same trick in eWLC releases before 17.3.1:

Copy the support bundle from AP via SCP to eWLC IP if you don't have a TFTP server reachable to the AP.

The eWLC is usually reachable via SSH from the AP, so that's a good trick for pre-17.3.

Step 1. [Enable SSH on 9800 v17.2.1](#)

Step 2. [Enable SCP on Cisco IOS® XE v17.2.1](#)

This example shows how to configure the server-side functionality of SCP. This example uses a locally defined username and password:

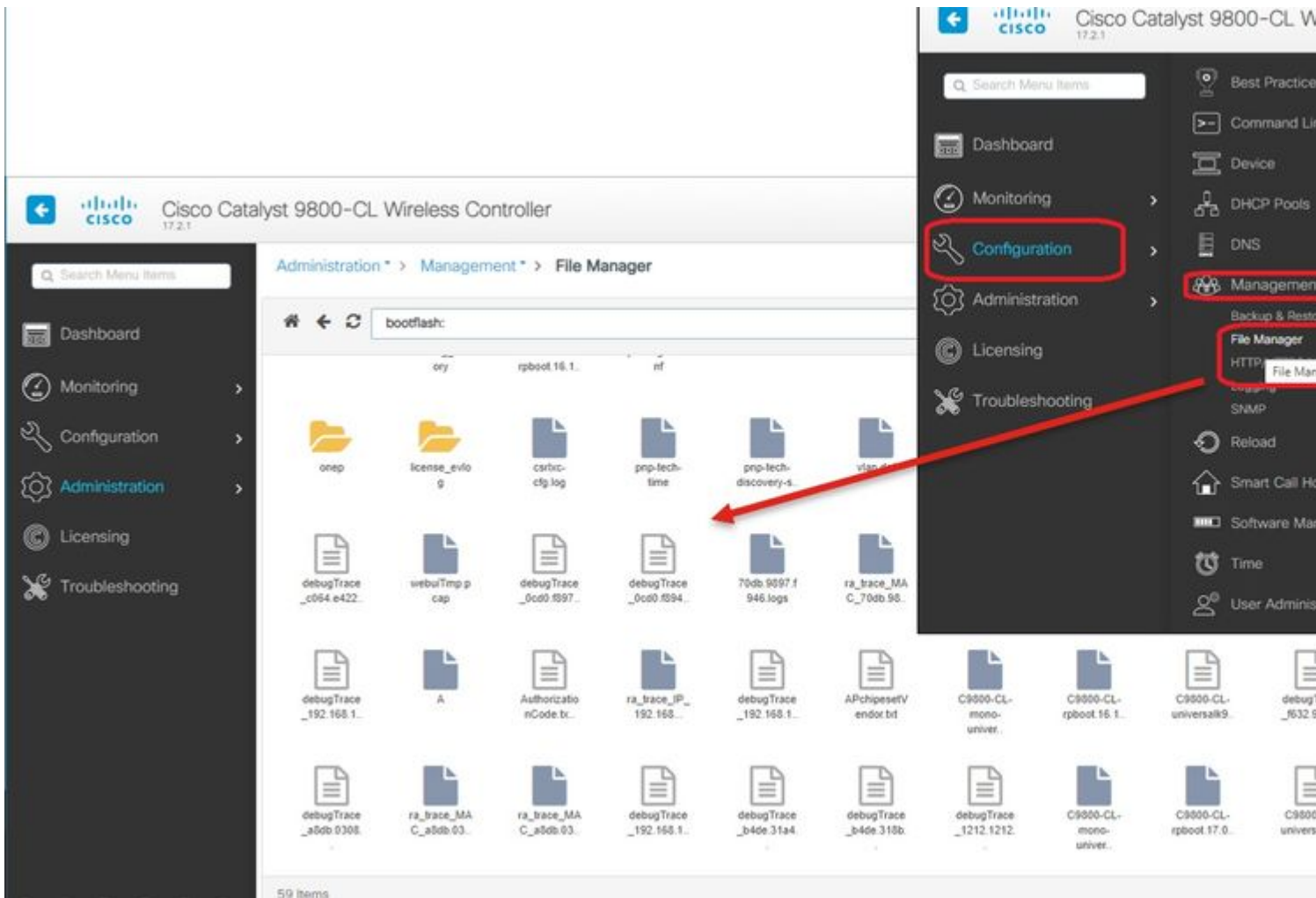
```
! AAA authentication and authorization must be configured properly in order for SCP to work.
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
Device(config)# ip scp server enable
Device(config)# end
```

Step 3. Use the command `copy support-bundle` and we need to specify the filename to be created in the SCP server.

Tip: You can run the command once to get a meaningful filename, and then copy/paste that filename in the command:

```
AP70DB.98E1.3DEC#copy support-bundle scp: admin@192.168.1.15:/
Creating support bundle, please wait...!tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110006.tgz created ===+
Warning: Permanently added '192.168.1.15' (RSA) to the list of known hosts.
Password:
Connection closed by 192.168.1.15 port 22
lost connection
AP70DB.98E1.3DEC#copy support-bundle scp: admin@192.168.1.15:/AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110006.tgz
Creating support bundle, please wait...!tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110400.tgz created ===+
Password:
AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110400.tgz
Connection to 192.168.1.15 closed by remote host.
AP70DB.98E1.3DEC#
```

Step 4. Then you can go into the eWLC GUI and get the file under: **Administration > Management > File Manager**:



IoT and Bluetooth

The gRPC server logs can be checked on the AP with :

```
AP# show grpc server log
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces conn url 10.22.243.33:8000"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering stopDNASpacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting stopDNASpacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering startDNASpacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] launching token request cycle"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting startDNASpacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces token expiration time 2020-04-02 01:36:52 +0000"
time="2020-04-01T01:36:52Z" level=info msg="Calling startDNASpacesConn routine "
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Receive Success status"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Connection not in ready state sleeping for 10 seconds"
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Setup Stream for the gRPC connection"
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Connect RPC Succeeded."
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] RX routine got enabled "
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] TX routine got enabled "
```

Connectivity to DNA Spaces connector can be verified with :

To see the scanned results :

```
AP# show controllers ioTRadio ble 0 scan brief
  Profile          MAC      RSSI(-dBm)  RSSI@1meter(-dBm)  Last-heard
Unknown 3C:1D:AF:62:EC:EC      88           0 0000D:00H:00M:01S
iBeacon 18:04:ED:04:1C:5F      86           65 0000D:00H:00M:01S
Unknown 18:04:ED:04:1C:5F      78           65 0000D:00H:00M:01S
Unknown 04:45:E5:28:8E:E7       85           65 0000D:00H:00M:01S
Unknown 2D:97:FA:0F:92:9A       91           65 0000D:00H:00M:01S
iBeacon E0:7D:EA:16:35:35      68           65 0000D:00H:00M:01S
Unknown E0:7D:EA:16:35:35      68           65 0000D:00H:00M:01S
iBeacon 04:EE:03:53:74:22     45          256 0000D:00H:00M:01S
Unknown 04:EE:03:53:74:22     45          256 0000D:00H:00M:01S
        04:EE:03:53:6A:3A     72           N/A 0000D:00H:00M:01S
Unknown 04:EE:03:53:6A:3A     72           65 0000D:00H:00M:01S
iBeacon E0:7D:EA:16:35:35      68           65 0000D:00H:00M:01S
Unknown E0:7D:EA:16:35:35      67           65 0000D:00H:00M:01S
iBeacon 04:EE:03:53:74:22     60          256 0000D:00H:00M:01S
Unknown 04:EE:03:53:74:22     60          256 0000D:00H:00M:01S
Eddystone URL 04:EE:03:53:6A:3A     72           N/A 0000D:00H:00M:01S
```

When the AP acts in Advanced BLE gateway mode where an app is deployed, you can check the status of the IoX application with :

```
AP#show iox applications
Total Number of Apps : 1
-----
App Name          : cisco_dnas_ble_iox_app
App Ip            : 192.168.11.2
App State         : RUNNING
App Token         : 02fb3e98-ac02-4356-95ba-c43e8a1f4217
App Protocol      : ble
App Grpc Connection : Up
Rx Pkts From App  : 3878345
Tx Pkts To App    : 6460
Tx Pkts To Wlc   : 0
Tx Data Pkts To DNASpaces : 3866864
Tx Cfg Resp To DNASpaces : 1
Rx KeepAlive from App : 11480
Dropped Pkts      : 0
App keepAlive Received On : Mar 24 05:56:49
```

You can connect to the IOX application with these commands and then monitor the logs during floor beacon configuration :

```
AP#connect iox application
/ #

/# tail -F /tmp/dnas_ble.log
Tue Mar 24 06:55:21 2020 [INFO]: Starting DNA Spaces BLE IOx Application
Tue Mar 24 06:55:21 2020 [INFO]: Auth token file contents: db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Setting gRPC endpoint to: 1.1.7.101:57777
```

```
Tue Mar 24 06:55:21 2020 [INFO]: Auth with token: db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Attempt to connect to DNAS Channel
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run metrics
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run Channel Keepalive
Tue Mar 24 06:55:21 2020 [INFO]: Initialize DNAS Reader Channel
Tue Mar 24 06:55:21 2020 [INFO]: Start listener for messages
Tue Mar 24 06:55:21 2020 [INFO]: Running BLE scan thread
```

Conclusion

There are many troubleshooting tools available to help us in the resolutions of problems related to COS APs.

This document lists the most commonly used ones and is regularly updated.