aruba
a Hewlett Packard
Enterprise company

IoT – Basic Setup Guide

Solution Guide

Copyright Information

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett Packard Enterprise Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett Packard Enterprise Company
Attn: General Counsel
3000 Hanover Street
Palo Alto, CA 94304
USA

aruba
a Hewlett Packard
Enterprise company

www.arubanetworks.com

3333 Scott Blvd

Santa Clara, CA 95054

Phone: 1-800-WIFI-LAN (+800-943-4526)

Fax 408.227.45

Contents

Revision History

The following table lists the revisions of this document:

Revision	Change Description
Revision 2	Added Zigbee configuration
Revision 1	Initial Publication

Intended Audience

This document is intended for enabling new partners to set up Aruba's Controller based and Instant based wireless network architecture to test IoT configurations.

This document can also be useful for SEs and customers who are looking to setup IoT devices supported by Aruba.

This document mainly aims at providing configuration and setup details for BLE and Zigbee transport profile mechanisms and majorly aims at AOS and Instant OS versions 8.6 and above.

Introduction

The Internet of Things (IoT) can be defined as a universe of devices, software, and systems that interact directly with the physical environment while communicating with each other and the IT infrastructure. Today it's almost impossible to read a technical journal, sometimes a daily paper, without some reference to the Internet of Things (IoT). The term IoT is now bandied about in so many different contexts that its meaning, and the power of the insights it represents, are often lost in the noise. Let's look at what Aruba is doing in IoT space.

Aruba's Involvement

IoT can change the experience we have interacting with our workspaces, people and even machines. By leveraging data – like temperature, speed, location, or applications in use – IoT solutions enabled by Aruba help deliver meaningful experiences using context-aware network generated content.

Some of the key areas where IoT can have major impact that Aruba has identified can be listed as: Smart Digital Workplaces, Industrial and Manufacturing areas, Healthcare, Retail and Logistics.

Aruba supports IoT applications based on Wi-Fi (e.g. Wi-Fi tracking), BLE and Zigbee by providing the connection layer using Aruba APs as gateways. IoT applications that leverage these protocols are enabled with the IoT transport profile mechanism. The feature on ArubaOS and Instant OS developed for this capability is called IoT transport profile and it comprises of attributes that describe the connection to the server where the IoT data is to be transported. In addition, there are attributes that can be set to send the IoT data with a finer granularity. Via the IoT transport profile, ArubaOS supports multiple transport mechanisms (HTTPS POSTs, WebSockets), payload encoding (JSON, Google Protocol buffer 2.0), device classes, periodicity of information updates, and so on.

Starting from ArubaOS 8.4.0.0, two standardized endpoints, Telemetry-HTTPS and Telemetry-websocket are provided. The message payload for these two standardized endpoints are constructed based on a published JSON schema and Google protocol buffer (protobuf) format respectively. These new endpoints provide the capability to select new device classes in the aforementioned IoT transport profile.

Server Type

ArubaOS and Instant supports following server types/endpoints:

- Telemetry-Https
- Telemetry-Websocket
- Assa-Abloy
- Meridian Asset Tracking
- Meridian Beacon Management
- ZF Openmatics

You can configure multiple device classes for these endpoints. Data from IoT devices that match the configured device class (es) are sent to the server specified in the IoT transport profile. This provides flexibility to choose the payload content along with the time interval and endpoints in the IoT transport profile.

Device Class

ArubaOS and Instant supports the following device classes/payload content:

- Aruba Managed Beacons
- Aruba Managed Tags
- EnOcean Switches
- EnOcean Sensors
- Eddystone
- Assa Abloy
- ZF Tags
- iBeacons
- Mysphera
- Wifi-assoc-sta
- Wifi-unassoc-sta
- All BLE Data
- Aruba Sensors
- Wifi-tags*
- sBeacons*

*Introduced in AOS and Instant OS 8.6 onwards

IoT and Third Party Server Integration

ArubaOS and Instant enables the integration of IoT data with third party servers.

This integration provides a flexible interface for end-users/system integrators to build their own independent IoT service.

Network Configuration

Read the following section to configure Aruba wireless setup.

AOS – Controller Based Setup

This is for basic setup with one standalone controller and a few APs which should be sufficient for testing IoT configuration on Aruba gear.

Initial Setup on a Serial Port Connection

The serial port is located on the front panel (back panel in case of 7024 and 7008 controllers) of the managed device. You can start the Initial Setup dialog when you connect a terminal, PC or workstation running a terminal emulation program to the serial port on the managed device.

The serial port connection only allows you to configure the basic configuration required to connect the managed device to the network. The recommended browser-based configuration Wizard allows you to also install software licenses (Will need to install AP and PEF license) and configure internal and guest WLANs. If you use the Initial Setup dialog to configure the managed device, the browser-based Setup Wizard will not be available unless you reset the managed device to its factory default configuration.

To run the Initial full setup dialog from a serial connection:

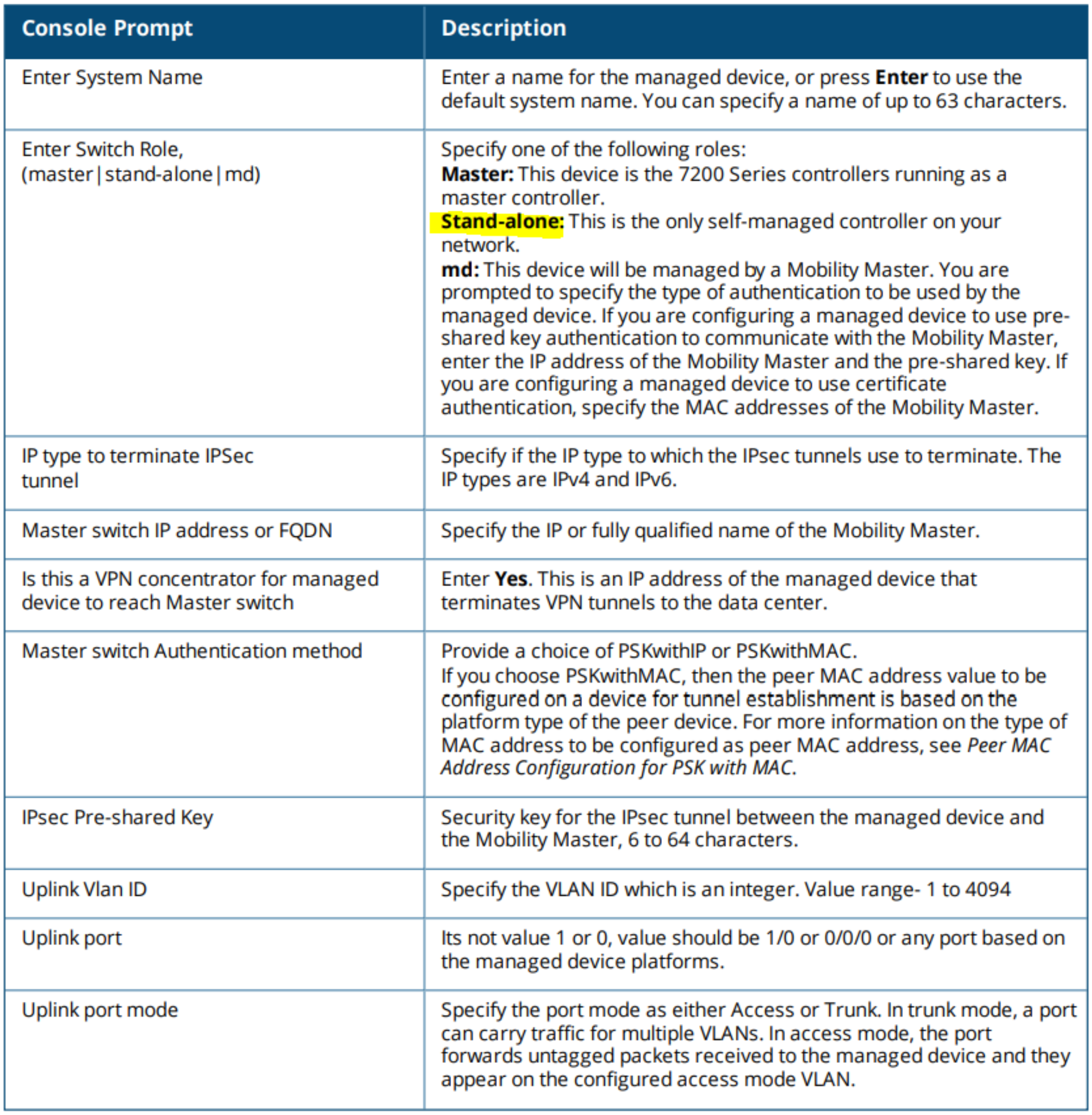
1. Connect your terminal or PC/workstation to the serial port on the managed devices using an RS-232 serial cable. RJ-45 cable and DB-9 to RJ-45 adapter is required. You may need a USB adapter to connect the serial cable to your PC.

2. Boot up the managed device. After the managed device has booted up, you should see a screen similar to the following setup dialog for managed devices:

```
Auto-provisioning is in progress. Choose one of the following options to override
or debug...

'enable-debug' : Enable auto-provisioning debug logs
'disable-debug': Disable auto-provisioning debug logs
'mini-setup'   : Stop auto-provisioning and start mini setup dialog for smart-
branch role
'full-setup'   : Stop auto-provisioning and start full setup dialog for any role


Enter Option (partial string is acceptable):f
Are you sure that you want to stop auto-provisioning and start full setup dialog?
(yes/no): y
Reading configuration from factory-default.cfg
```

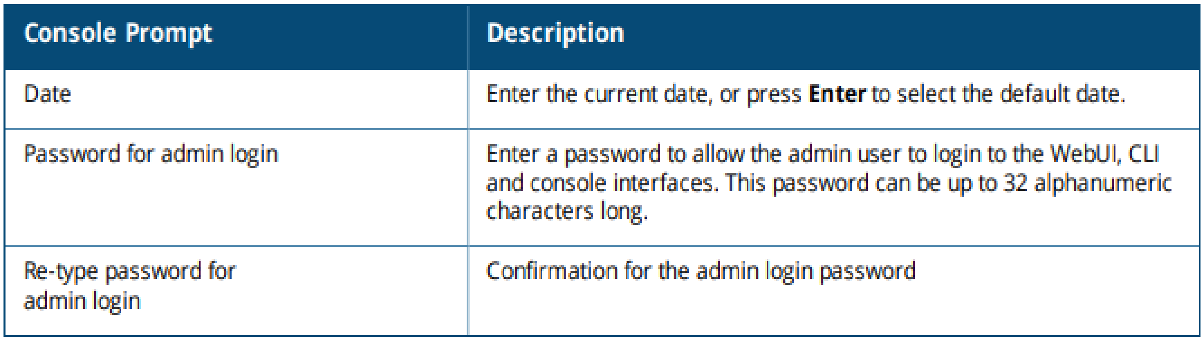
3. Enter f to invoke the full-setup.

4. The Serial Port Configuration Dialog displays the configuration prompts. The prompts may vary, depending upon the switch role you choose (For this setup, use ‘Stand-alone’). Enter the required information at each prompt, then press Enter to continue to the next question



Console Prompt	Description
Enter System Name	Enter a name for the managed device, or press **Enter** to use the default system name. You can specify a name of up to 63 characters.
Enter Switch Role, (master|stand-alone|md)	Specify one of the following roles: **Master:** This device is the 7200 Series controllers running as a master controller. **Stand-alone:** This is the only self-managed controller on your network. **md:** This device will be managed by a Mobility Master. You are prompted to specify the type of authentication to be used by the managed device. If you are configuring a managed device to use pre-shared key authentication to communicate with the Mobility Master, enter the IP address of the Mobility Master and the pre-shared key. If you are configuring a managed device to use certificate authentication, specify the MAC addresses of the Mobility Master.
IP type to terminate IPSec tunnel	Specify if the IP type to which the IPsec tunnels use to terminate. The IP types are IPv4 and IPv6.
Master switch IP address or FQDN	Specify the IP or fully qualified name of the Mobility Master.
Is this a VPN concentrator for managed device to reach Master switch	Enter **Yes**. This is an IP address of the managed device that terminates VPN tunnels to the data center.
Master switch Authentication method	Provide a choice of PSKwithIP or PSKwithMAC. If you choose PSKwithMAC, then the peer MAC address value to be configured on a device for tunnel establishment is based on the platform type of the peer device. For more information on the type of MAC address to be configured as peer MAC address, see *Peer MAC Address Configuration for PSK with MAC.*
IPsec Pre-shared Key	Security key for the IPsec tunnel between the managed device and the Mobility Master, 6 to 64 characters.
Uplink Vlan ID	Specify the VLAN ID which is an integer. Value range- 1 to 4094
Uplink port	Its not value 1 or 0, value should be 1/0 or 0/0/0 or any port based on the managed device platforms.
Uplink port mode	Specify the port mode as either Access or Trunk. In trunk mode, a port can carry traffic for multiple VLANs. In access mode, the port forwards untagged packets received to the managed device and they appear on the configured access mode VLAN.

Console Prompt	Description
Enter Native VLAN ID [1]	Specify a particular vlan to be configured as a native vlan.
Uplink Vlan IP assignment method	Assign manually the IP addressing of the uplink or via DHCP.
Uplink Vlan Static IP address	The managed device takes its IP address from VLAN 1 and uses this IP address to communicate with other managed devices and with APs. Enter an IPv4 VLAN 1 interface IP address, or press **Enter** without specifying an IP address to use the default address 172.16.0.254/24.
Uplink Vlan Static IP netmask	Enter an IPv4 VLAN 1 interface IP subnet mask, or press Enter without specifying an IP address to use the default address 255.255.255.0.
IP default gateway	This is usually the IP address of the interface on the upstream switch or router to which you will connect the managed devices. The default gateway and the VLAN 1 IP address need to be in the same network. Enter an IPv4 gateway IP address, or press Enter to continue without specifying an IP gateway.
DNS IP address	IP address of the DNS server.
IPV6 address on vlan	IPv6 address of the managed device.
Do you want to configure port-channel (yes|no) [no]	Specify if you want to configure the port-channel. LACP will be configured on port members with port-channel ID as LACP group ID.
Enter Port-channel ID [0]	Specify the port-channel ID.
Uplink Vlan Static IPv6 address	The managed device takes its IP address from VLAN 1 and uses this IP address to communicate with other managed devices and with APs. Supported subnets are: Global Unicast: 2000::/3, Unique local unicast: fc00::/7 Enter an IPv6 VLAN 1 interface IP address, or press **Enter** without specifying an IP address to use the default address 2000::1.
Uplink Vlan interface IPV6 prefix length	Enter a value from 0 to 128 to define an IPv6 VLAN 1 interface IP prefix length, or press **Enter** without specifying a prefix length to use the default value of 64.
IPv6 default gateway	This optional value is usually the IP address of the interface on the upstream switch or router to which you will connect the managed device. The default gateway and the VLAN 1 IP address need to be in the same network. Enter an IPv6 gateway IP address to configure this setting, or press **Enter** to continue without specifying an IP gateway.
Country code	If your managed device has a country code that restricts its usage, enter **yes** to confirm this code.
Time Zone	Enter the time zone for the managed device, or press **Enter** to select the default time zone.
Time in UTC	Enter the current time in UTC format, or press **Enter** to select the default time.



Console Prompt	Description
Date	Enter the current date, or press **Enter** to select the default date.
Password for admin login	Enter a password to allow the admin user to login to the WebUI, CLI and console interfaces. This password can be up to 32 alphanumeric characters long.
Re-type password for admin login	Confirmation for the admin login password

Install Licenses

Contact local SEs or TAC to obtain license keys.

Use the following procedure to install licenses:

1. Enter the IP address of the Mobility Controller in the URL of a browser window to access the WebUI.

2. At the WebUI login page, enter the admin user name and the password you entered during the Initial Setup.

3. Click on ‘Mobility Controller’ hierarchy node on top left.

4. Navigate to the Configuration > License > Inventory window.

5. Click on the + sign to add new license keys.

6. Navigate to Configuration > License > Usage. Check all the required features (AP and PEF) next to ‘Feature Enabled’.

7. Click Submit.

8. Click Pending Changes.

9. In the Pending Changes window, select the check box and click Deploy changes.

Configure the Mobility Controller to Support APs

Before you install APs in a network environment, you must ensure that the APs will be able to locate and connect to the managed device when powered on. Specifically, you need to ensure the following:

- When connected to the network, each AP is assigned a valid IP address
- APs are able to locate the managed devices

Each Aruba AP requires a unique IP address on a subnetwork that has connectivity to a managed device. Aruba recommends using the DHCP to provide IP addresses for APs; the DHCP server can be an existing network server or an Aruba managed device configured as a DHCP server.

If an AP is on the same subnetwork as the Mobility Controller, you can configure the Mobility Controller as a DHCP server to assign an IP address to the AP. The Mobility Controller must be the only DHCP server for this subnetwork.

Enable DHCP Server Capability

Use the following procedure to use the WebUI to enable DHCP server capability:

1. Enter the IP address of the Mobility Controller in the URL of a browser window to access the WebUI.

2. At the WebUI login page, enter the admin user name and the password you entered during the Initial Setup.

3. Navigate to the Configuration > Services window.

4. Open the DHCP Server tab.

5. Select Enable from either IPv4 or IPv6 DHCP server drop-down list.

6. In the Pool Configuration table, click +.

7. Enter information about the subnetwork for which IP addresses are to be assigned.

8. Click Submit.

9. If there are addresses that should not be assigned in the subnetwork:

    a. Click + in the Excluded Address Range section.

    b. Enter the address range in the Add Excluded Address section.

    c. Click Submit.

10. Click Pending Changes.

11. In the Pending Changes window, select the check box and click Deploy changes.

Mobility Controller Discovery

An Aruba AP can discover the IP address of the Mobility Controller in one of several ways. The ADP is enabled by default on all Aruba APs and Mobility Controller. If all APs and Mobility Controller are connected to the same Layer-2 network, APs will use ADP to discover their Mobility Controller. If the devices are on different networks, you must configure the AP to use a Layer-3 compatible discovery mechanism such as DNS, DHCP, or IGMP forwarding after installing the AP on the network. For details, refer to the ArubaOS 8.5.0.x User Guide.

With ADP, APs send out periodic multicast and broadcast queries to locate the controller. If the APs are in the same broadcast domain as the Mobility Controller, the Mobility Controller automatically responds to the APs' queries with its IP address. If the APs are not in the same broadcast domain as the Mobility Controller, you need to enable multicast on the network. If multicast is not an option, then the APs can be configured to use DNS or DHCP based provisioning to contact the managed device.

Install the Access Points

You can either connect the AP directly to a port on the managed device, or connect the AP to another switch or router that has Layer-2 or Layer-3 connectivity to the managed device. If the Ethernet port on the managed device is an 802.3af PoE port, the AP automatically uses it to power up. If a PoE port is not available, contact your Aruba vendor to obtain an AC adapter for the AP.

Once an AP is connected to the network and powered up, it will automatically attempt to locate the Mobility Controller. You can view a list of all APs connected to the Mobility Controller by accessing the Configuration > Access Points page in the WebUI of the Mobility Controller. An AP installed on the network advertises its default SSID. Wireless users can connect to this SSID, but will not have access to the network until you configure authentication policies and user roles for your wireless users. For complete details on authentication policies and user roles, refer to the ArubaOS 8.5.0.x User Guide.

NOTE

For any additional information or help regarding installation, please check the user guide click here.

Upgrading Controller Firmware

Download the latest image file from Support website here:

Click Here

Once logged in, 'Download Software' tab will be seen. Appropriate file needs to be downloaded depending on the Controller model.

Once downloaded:

- Login into the UI with username and password set above.
- Go to Maintenance > Software Management> Upgrade > Upgrade using=Local File >Browse> Select the Image file downloaded from the support website> Select the partition> Upgrade

Instant

Before installing an Instant AP (IAP):

- Ensure that you have an Ethernet cable of the required length to connect an Instant AP to the home router.
- Ensure that you have one of the following power sources:
  - IEEE 802.3af/at-compliant PoE source. The PoE source can be any power source equipment switch or midspan power source equipment device.
  - Instant AP power adapter kit.

Connecting an Instant AP

Based on the type of the power source used, perform one of the following steps to connect an Instant AP to the power source:

- PoE switch—Connect the Ethernet 0 port of the Instant AP to the appropriate port on the PoE switch.
- PoE midspan—Connect the Ethernet 0 port of the Instant AP to the appropriate port on the PoE midspan.
- AC to DC power adapter—Connect the 12V DC power jack socket to the AC to DC power adapter

Assigning an IP address to the Instant AP

The Instant AP needs an IP address for network connectivity. When you connect an Instant AP to a network, it receives an IP address from a DHCP server.

To obtain an IP address for an Instant AP:
1. Ensure that the DHCP service is enabled on the network.
2. Connect the Ethernet 0 port of Instant AP to a switch or router using an Ethernet cable.
3. Connect the Instant AP to a power source. The Instant AP receives an IP address provided by the switch or router.

You will be able to view the IP address assigned to the AP in the UI later or issue “show ip int brief” if you have the console access.

Assigning a Static IP

To assign a static IP to an Instant AP:

1. Connect a terminal, PC, or workstation running a terminal emulation program to the Console port on the Instant AP.
2. Turn on the Instant AP. An autoboot countdown prompt that allows you to interrupt the normal startup process and access apboot is displayed.
3. Press Enter key before the timer expires. The Instant AP goes into the apboot mode.
4. In the apboot mode, execute the following commands to assign a static IP to the Instant AP.

```
Hit <Enter> to stop autoboot: 0
apboot> factory_reset
apboot> setenv ipaddr 192.0.2.0
apboot> setenv netmask 255.255.255.0
apboot> setenv gatewayip 192.0.2.2
apboot> save
Saving Environment to Flash... Un-Protected 1 sectors .done Erased 1 sectors Writ-
ing
```

5. Use the printenv command to view the configuration.

```
apboot> printenv
apboot> boot
```

The AP will reboot and once it has completely rebooted, it will start broadcasting default SSID: SetMeUp-xx:xx:xx. At this stage the Instant AP is up and running. You can now connect to this SSID to do further configuration through UI (Explained in next step: Connecting to a Provisioning Wi-Fi Network).

At this point, you can start configuring the IAP for IoT through CLI.

NOTE

Aruba Instant APs do not require licenses.

NOTE

All the subsequent IAPs will join this 1st IAP if they are brought up in the same subnet and will form a cluster.

Connecting to a Provisioning Wi-Fi Network

To connect to a provisioning Wi-Fi network:

1. Ensure that the client/endpoint is not connected to any wired network.

2. Connect a wireless-enabled client to a provisioning Wi-Fi network: for example, SetMeUp-xx:xx:xx.

3. Launch a web browser and enter http://setmeup.arubanetworks.com or IP address of the AP.

4. In the login screen, enter the following credentials:

- Username—admin
- Password—admin (For Instant OS 8.5 onwards, default password is the serial number of the AP)

5. Once logged in, new SSIDs and other configuration can be created.

Upgrading AP Firmware

Download the latest image file from Support website here:

Click Here

Once logged in, 'Download Software' tab will be seen.

Appropriate file needs to be downloaded depending on the AP model:

AP Model	Image Name
RAP 155, RAP-155P	Aries
214/215, 224/225, 274/275, 277	Centaurus
344/345, 514/515	Draco
314/315, 324/325, 318, 374/375/377/387	Hercules
334/335	Lupus
534/535, 555	Scorpio
303/303P,303H, 304/305, 365/367	Ursa
203H, 203R, 203RP, 207	Vela
504,505	Gemini

Once downloaded:

- Login into the UI with username and password set above.
- Go to Maintenance > Firmware> Manual > Select 'Image File' >Browse> Select the Image file downloaded from the support website> Upgrade Now

Configuring IoT

BLE

This section covers IoT configuration for BLE on AOS and Instant. It covers the steps for enabling BLE radio, radio profile and transport profile. It also covers some helpful monitoring commands and finally goes through sample configurations and show command output.

AOS

Enable BLE

For AOS versions prior to 8.6

```
(host) [mynode]# configure terminal
(host) [mynode] (config)# ap system-profile <default>
(host) [mynode] (config)# ble-op-mode beaconing
```

For AOS versions 8.6 and above

```
(host) [mynode] (config)# iot radio-profile Test
(host) [mynode] (IoT Radio Profile "test")# radio-mode ble
```

Radio Profile Configuration Options

Config Option	Description
`ble-console`	BLE console mode
`ble-opmode`	BLE operational mode
`ble-txpower`	BLE transmit power in dBm
`clone`	Copy data from another IoT Radio Profile
`no`	Delete Command
`radio-instance`	IoT radio instance
`radio-mode`	IoT radio modes
`zigbee-channel`	ZigBee scanning channel[auto, 11-26]
`zigbee-opmode`	ZigBee operational mode

In AOS 8.6 onwards, radio-instance is set to ‘internal’ by default meaning internal antennas will be used. Ble-opmode is set to ‘beaconing+scanning’ by default. In AOS 8.5 and prior, ‘ble-opmode beaconing’ was used to enable both, beaconing and scanning at the same time.

Enable the IoT Radio Profile (For AOS 8.6 and above)

```
(host) [mynode](IoT Radio Profile "test")#ap-group <default>
(host) [mynode](AP group "default") #iot radio-profile Test
```

Configure IoT Transport Profile

```
(host) [mynode] (config) #iot transportProfile Sample-Websocket
(host) [mynode] (IoT Data Profile "Sample-Websocket ") #serverType Telemetry-Websocket
(host) [mynode] (IoT Data Profile "Sample-Websocket ") #deviceClassFilter sbeacon
(host) [mynode] (IoT Data Profile "Sample-Websocket ") #serverURL <URL>
(host) [mynode] (IoT Data Profile "Sample-Websocket ") #reportingInterval <time in seconds>
(host) [mynode] (IoT Data Profile "Sample-Websocket ") #include-ap-group default
```

NOTE

Depending on the server side configuration and requirement, if you are using ‘`authenticationURL`’ instead of ‘`serverURL`’, you will need to provide ‘username’ & ‘password’ or ‘clientID’ & ‘accessToken’.

NOTE

ReportingInterval: Default=600s; Min for telemetry websocket=1s; Min for telemetry https=5s.

NOTE

There are other parameters inside transportProfile mentioned in the below table which also can be edited as per requirement. However, above mentioned steps can get you started with minimum configuration.

Transport Profile Configuration Options

Config Option	Short Description	Long Description
`serverType`	Server Type	The type of server that is receiving the telemetry stream
`authenticationURL`	Authentication URL	Server URL for authentication
`username`	Username	Username for authentication
`password`	Password	Password for authentication
`clientID`	Client ID	This ID identifies the sender to the server
`serverURL`	Server URL	Server URL for sending telemetry
`accessToken`	Access Token	Access token. Configure this only if you want to bypass authentication
`reportingInterval`	Reporting Interval	Reporting interval in seconds

`deviceClassFilter`	Device Class Filter	A list of device class tags to filter the devices included in the reports
`uuidFilter`	UUID Filter	A list of UUIDs to filter the devices included in the reports. Applies only to iBeacon devices
`uidNamespaceFilter`	UID namespace Filter	A list of UID namespaces to filter devices included in the reports. Applies only Eddystone-UID devices
`urlFilter`	URL Filter	A list of URL strings to filter devices included in the reports. Applies only Eddystone-URL devices. The string listed here can be partial URL strings
`cellSizeFilter`	Cell Size Filter	A proximity filter. Devices outside the cell will not be reported. Size is specified in meters. Setting to 0 disables the cell size filter
`movementFilter`	Movement Filter	Filters devices that do not change distance. Specified in meters. Applicable only if a cell size is set. Setting to 0 disables the movement filter
`ageFilter`	Age Filter	Age filter. Devices without recent activity will not be reported
`rssiReporting`	RSSI Reporting Format	Set the preferred format for RSSI reporting

Enable the IoT Transport Profile

To apply an IoT transport profile, execute the following command:

```
(host) [mynode] (config) #iot useTransportProfile Sample-Websocket
```

Useful command: # `ble_relay send_sync_iotcfg`. This will force sync the transportProfile config with the APs.

Show Commands

To view the list of IoT transport profiles, execute the following command:

```
(host) [mynode] (config) #show iot transportProfile
```

To view the status of an IoT transport profile, execute the following command:

```
(host) [mynode] (config) #show iot transportProfile test-IoT
```

Use the following command to view any third-party devices in the BLE table:

```
(host) [mynode] (config) # show ap debug ble-table ap-name <ap_name> generic
```

```
(host) [mynode] (config)# show ap debug ble-table ap-name <ap_name> generic |
include enocean/ibeacon/sbeacon
```

To view the list of IoT radio profiles, execute the following command:

```
(host) [mynode] (config) #show iot radio-profile
```

To view the status of an IoT radio profile, execute the following command:

```
(host) [mynode] (config) #show iot radio-profile Test
```

Viewing BLE Status

To view the BLE relay status, execute the following command (This command shows that whether connection between the Aruba AP and the server has been established, what is the Transport type, and time details):

```
(host) [mynode] #show ble_relay iot-profile
```

To view the BLE configuration of an AP, execute the following command:

```
(host) [mynode] #show ap debug ble-config ap-name <ap_name>
```

To view the BLE relay report, execute the following command:

```
(host) [mynode] #show ble_relay report <profile-name>
```

To view the BLE relay jobs, execute the following command:

```
(host) [mynode] #show ble_relay jobs
```

To view the BLE relay status for websockets transport, execute the following command:

```
(host) [mynode] #show ble_relay disp-attr all
```

To view list of BLE relay tag report, execute the following command:

```
(host) [mynode] #show ble_relay tag-report <profile-name>
```

To view list of BLE relay WS log, execute the following command (This shows the detailed web socket log activity):

```
(host) [mynode] #show ble_relay ws-log <profile-name>
```

To view third-party devices in the BLE table, execute the following command:

```
(host) [mynode] #show ap debug ble-table ap-name <ap_name> generic
```

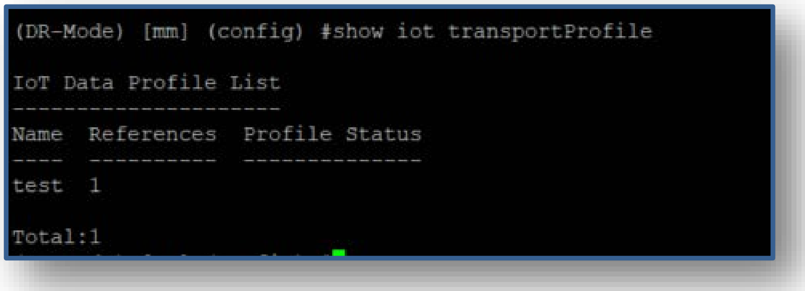
Sample Configuration and Output for AOS 8.6

Configuration

```
(host) [mynode] # conf t
(host) [mynode] (config) #iot transportProfile test
(host) [mynode] (IoT Data Profile "test") #serverType Telemetry-Websocket
(host) [mynode] (IoT Data Profile "test") #deviceClassFilter enocean-switches
(host) [mynode] (IoT Data Profile "test") #serverURL
ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
(host) [mynode] (IoT Data Profile "test") #reportingInterval 5
(host) [mynode] (IoT Data Profile "test") #include-ap-group default
(host) [mynode] (IoT Data Profile "test")# exit

(host) [mynode] (config) #iot useTransportProfile test

(host) [mynode] (config)# iot radio-profile test
(host) [mynode] (IoT Radio Profile "test")# radio-mode ble


(host) [mynode](IoT Radio Profile "test")#ap-group default
(host) [mynode](AP group "default") #iot radio-profile test
(host) [mynode](AP group "default") #end
(host) [mynode] #write memory
```

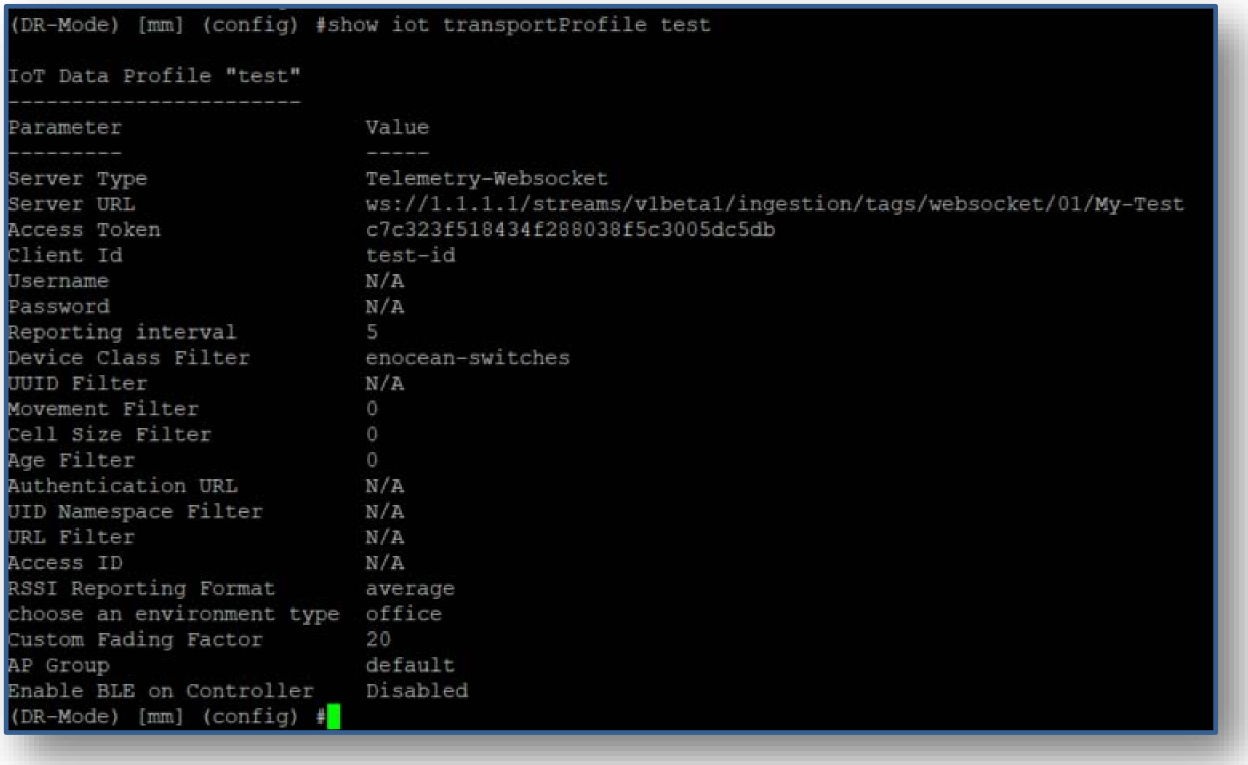
Output

#show iot transportProfile



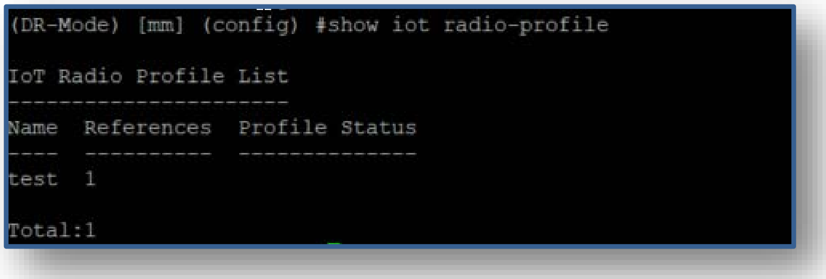
```
(DR-Mode) [mm] (config) #show iot transportProfile

IoT Data Profile List
---------------------
Name  References  Profile Status
----  ----------  --------------
test  1

Total:1
```

show iot transportProfile test



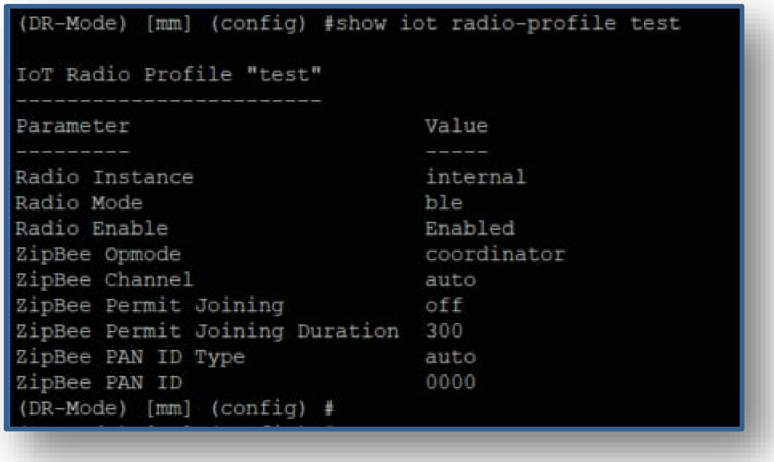
```
(DR-Mode) [mm] (config) #show iot transportProfile test

IoT Data Profile "test"
-----------------------
Parameter                   Value
---------                   -----
Server Type                 Telemetry-Websocket
Server URL                  ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
Access Token                c7c323f518434f288038f5c3005dc5db
Client Id                   test-id
Username                    N/A
Password                    N/A
Reporting interval          5
Device Class Filter         enocean-switches
UUID Filter                 N/A
Movement Filter             0
Cell Size Filter            0
Age Filter                  0
Authentication URL          N/A
UID Namespace Filter        N/A
URL Filter                  N/A
Access ID                   N/A
RSSI Reporting Format       average
choose an environment type  office
Custom Fading Factor        20
AP Group                    default
Enable BLE on Controller    Disabled
(DR-Mode) [mm] (config) #
```

#show iot radio-profile



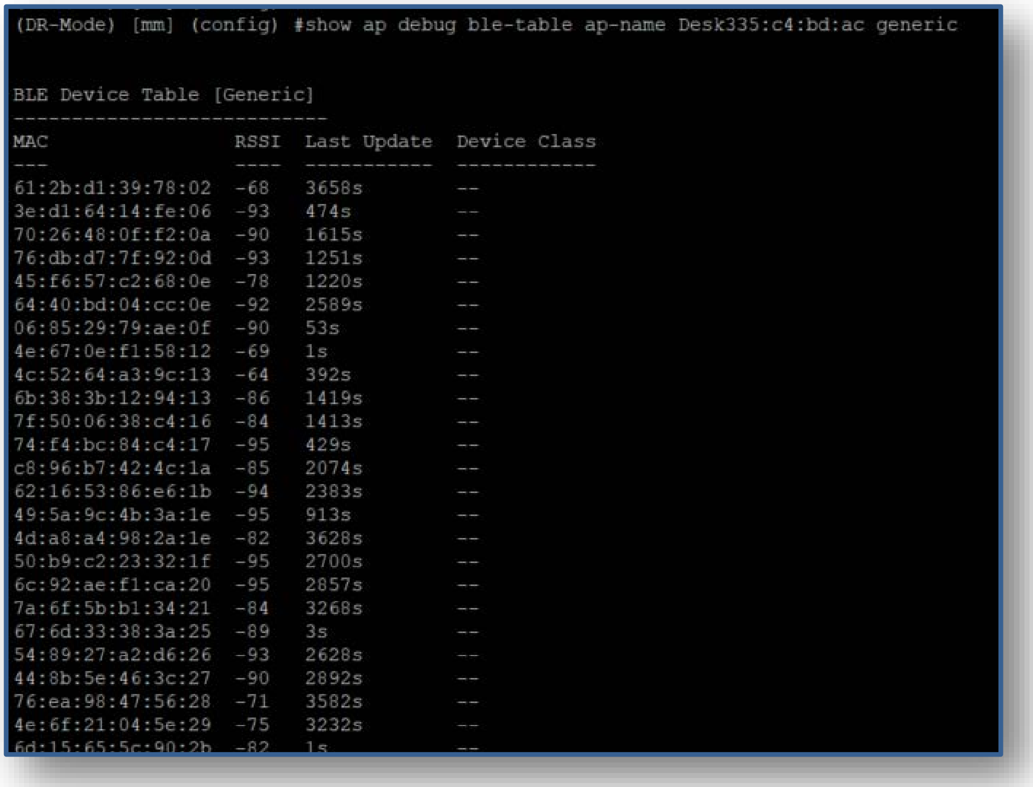
```
(DR-Mode) [mm] (config) #show iot radio-profile

IoT Radio Profile List
----------------------
Name  References  Profile Status
----  ----------  --------------
test  1

Total:1
```

#show iot radio-profile test



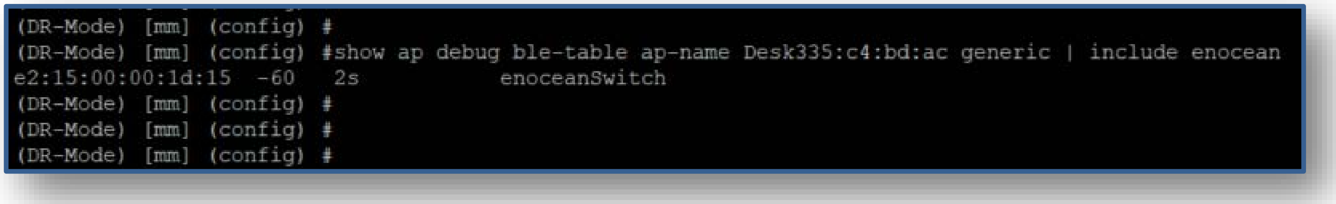
```
(DR-Mode) [mm] (config) #show iot radio-profile test

IoT Radio Profile "test"
------------------------
Parameter                         Value
---------                         -----
Radio Instance                    internal
Radio Mode                        ble
Radio Enable                      Enabled
ZipBee Opmode                     coordinator
ZipBee Channel                    auto
ZipBee Permit Joining             off
ZipBee Permit Joining Duration    300
ZipBee PAN ID Type                auto
ZipBee PAN ID                     0000
(DR-Mode) [mm] (config) #
```

#show ap debug ble-table ap-name Desk335:c4:bd:ac generic



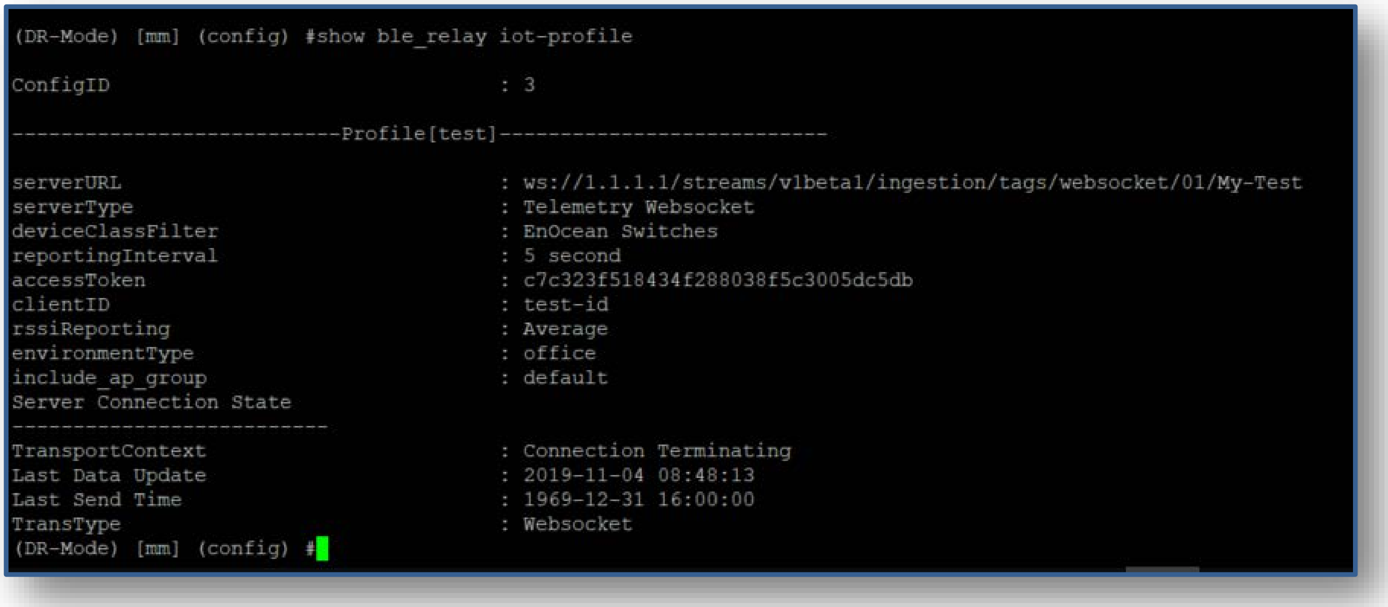
```
(DR-Mode) [mm] (config) #show ap debug ble-table ap-name Desk335:c4:bd:ac generic


BLE Device Table [Generic]
---------------------------
MAC                RSSI  Last Update  Device Class
---                ----  -----------  ------------
61:2b:d1:39:78:02  -68   3658s        --
3e:d1:64:14:fe:06  -93   474s         --
70:26:48:0f:f2:0a  -90   1615s        --
76:db:d7:7f:92:0d  -93   1251s        --
45:f6:57:c2:68:0e  -78   1220s        --
64:40:bd:04:cc:0e  -92   2589s        --
06:85:29:79:ae:0f  -90   53s          --
4e:67:0e:f1:58:12  -69   1s           --
4c:52:64:a3:9c:13  -64   392s         --
6b:38:3b:12:94:13  -86   1419s        --
7f:50:06:38:c4:16  -84   1413s        --
74:f4:bc:84:c4:17  -95   429s         --
c8:96:b7:42:4c:1a  -85   2074s        --
62:16:53:86:e6:1b  -94   2383s        --
49:5a:9c:4b:3a:1e  -95   913s         --
4d:a8:a4:98:2a:1e  -82   3628s        --
50:b9:c2:23:32:1f  -95   2700s        --
6c:92:ae:f1:ca:20  -95   2857s        --
7a:6f:5b:b1:34:21  -84   3268s        --
67:6d:33:38:3a:25  -89   3s           --
54:89:27:a2:d6:26  -93   2628s        --
44:8b:5e:46:3c:27  -90   2892s        --
76:ea:98:47:56:28  -71   3582s        --
4e:6f:21:04:5e:29  -75   3232s        --
6d:15:65:5c:90:2b  -82   1s           --
```

#show ap debug ble-table ap-name Desk335:c4:bd:ac generic | include enocean



```
(DR-Mode) [mm] (config) #
(DR-Mode) [mm] (config) #show ap debug ble-table ap-name Desk335:c4:bd:ac generic | include enocean
e2:15:00:00:1d:15  -60   2s           enoceanSwitch
(DR-Mode) [mm] (config) #
(DR-Mode) [mm] (config) #
(DR-Mode) [mm] (config) #
```

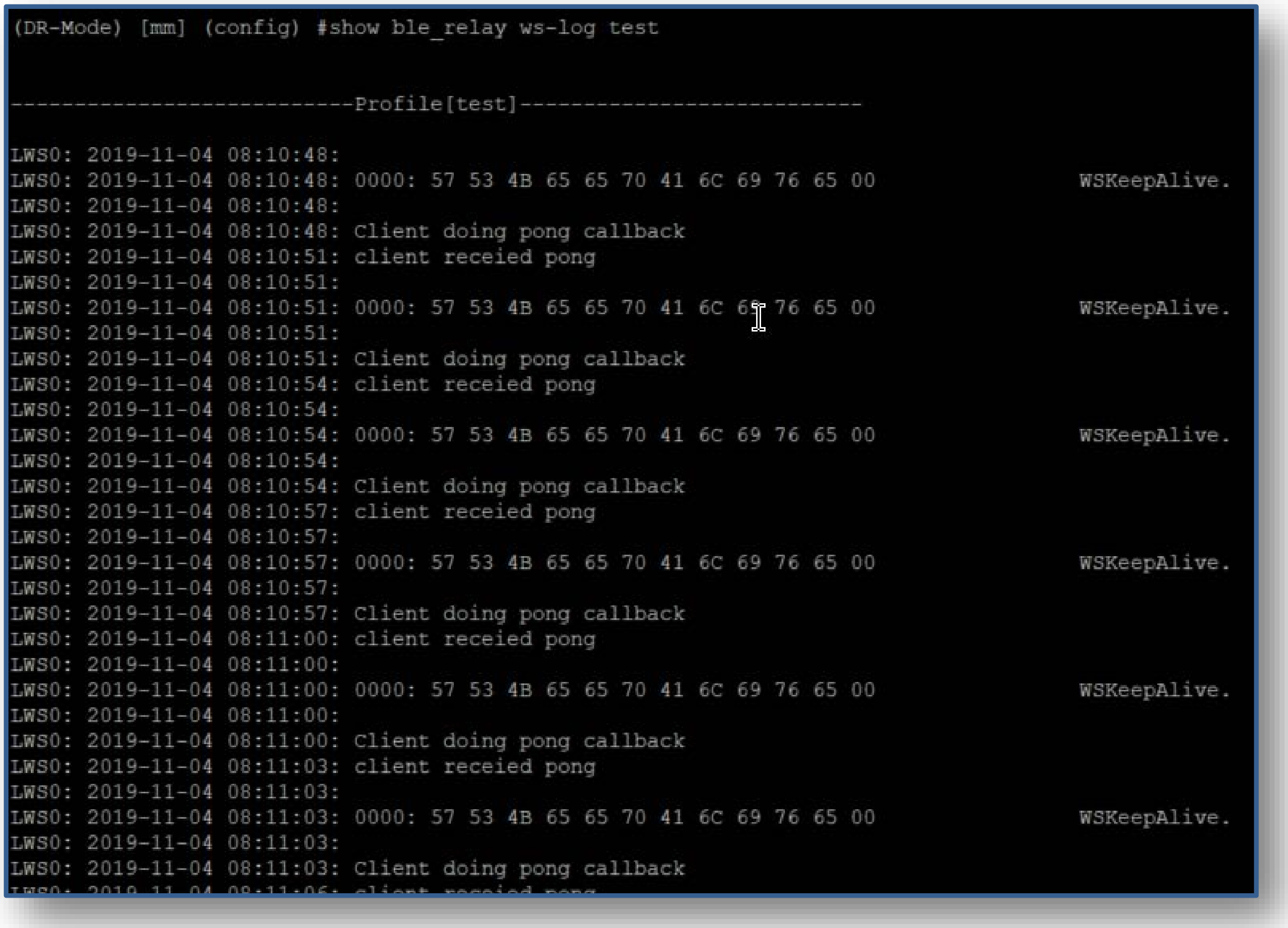
#show ble_relay iot-profile



```
(DR-Mode) [mm] (config) #show ble_relay iot-profile

ConfigID                                : 3

---------------------------Profile[test]---------------------------

serverURL                               : ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
serverType                              : Telemetry Websocket
deviceClassFilter                       : EnOcean Switches
reportingInterval                       : 5 second
accessToken                             : c7c323f518434f288038f5c3005dc5db
clientID                                : test-id
rssiReporting                           : Average
environmentType                         : office
include_ap_group                        : default
Server Connection State
--------------------------
TransportContext                        : Connection Terminating
Last Data Update                        : 2019-11-04 08:48:13
Last Send Time                          : 1969-12-31 16:00:00
TransType                               : Websocket
(DR-Mode) [mm] (config) #
```

*Once the connection is established, you should see “**TransportContext: Connection Established**”. This is the indicator that successful communication is happening between Aruba controller and the Web socket server

#show ble_relay ws-log test



```
(DR-Mode) [mm] (config) #show ble_relay ws-log test


---------------------------Profile[test]---------------------------

LWS0: 2019-11-04 08:10:48:
LWS0: 2019-11-04 08:10:48: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:10:48:
LWS0: 2019-11-04 08:10:48: Client doing pong callback
LWS0: 2019-11-04 08:10:51: client receied pong
LWS0: 2019-11-04 08:10:51:
LWS0: 2019-11-04 08:10:51: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:10:51:
LWS0: 2019-11-04 08:10:51: Client doing pong callback
LWS0: 2019-11-04 08:10:54: client receied pong
LWS0: 2019-11-04 08:10:54:
LWS0: 2019-11-04 08:10:54: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:10:54:
LWS0: 2019-11-04 08:10:54: Client doing pong callback
LWS0: 2019-11-04 08:10:57: client receied pong
LWS0: 2019-11-04 08:10:57:
LWS0: 2019-11-04 08:10:57: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:10:57:
LWS0: 2019-11-04 08:10:57: Client doing pong callback
LWS0: 2019-11-04 08:11:00: client receied pong
LWS0: 2019-11-04 08:11:00:
LWS0: 2019-11-04 08:11:00: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:11:00:
LWS0: 2019-11-04 08:11:00: Client doing pong callback
LWS0: 2019-11-04 08:11:03: client receied pong
LWS0: 2019-11-04 08:11:03:
LWS0: 2019-11-04 08:11:03: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-04 08:11:03:
LWS0: 2019-11-04 08:11:03: Client doing pong callback
LWS0: 2019-11-04 08:11:06: client receied pong
```

Instant

Enable BLE

For Instant OS versions prior to 8.6

```
(Instant AP) # configure terminal
(Instant AP) (config) # ble mode beaconing
```

For Instant OS versions 8.6 and above

```
(Instant AP) # configure terminal
(Instant AP)(config)# iot radio-profile test
(Instant AP) (IoT Radio Profile "test") #radio-mode ble
(Instant AP) (IoT Radio Profile "test") #exit
```

Enable the IoT Radio Profile (For Instant OS 8.6 and above)

```
(Instant AP)(config)# iot use-radio-profile test
```

Configure IoT Transport Profile

```
(Instant AP)(config)# iot transportProfile test
(Instant AP)(IoT Transport Profile "test")# endpointType telemetry-websocket
(Instant AP)(IoT Transport Profile "test")# payloadContent sbeacon
(Instant AP)(IoT Transport Profile "test")# endpointURL <url>
(Instant AP)(IoT Transport Profile "test")# transportInterval <time in seconds>
(Instant AP)(IoT Transport Profile "test")# end
(Instant AP) # commit apply
```

NOTE

Depending on the server side configuration and requirement, if you are using ‘authenticationURL’ instead of ‘endpointURL’, you will need to provide ‘username’ & ‘password’ or ‘EndpointToken’ & ‘EndpointID’.

NOTE

transportInterval: Default=600s; Min for telemetry websocket=1s; Min for telemetry https=5s.

NOTE

There are other parameters inside transportProfile mentioned in the below table which also can be edited as per requirement. However, above mentioned steps can get you started with minimum configuration.

Transport Profile Configuration Options

Config Option	Short Description	Long Description
`endpointType`	Server Type	The type of server that is receiving the telemetry stream
`authenticationURL`	Authentication URL	Server URL for authentication
`username`	Username	Username for authentication
`password`	Password	Password for authentication
`endpointID`	Client ID	This ID identifies the sender to the server
`endpointURL`	Server URL	Server URL for sending telemetry
`endpointToken`	Access Token	Access token. Configure this only if you want to bypass authentication
`transportInterval`	Reporting Interval	Reporting interval in seconds
`payloadContent`	Device Class Filter	A list of device class tags to filter the devices included in the reports
`uuidFilter`	UUID Filter	A list of UUIDs to filter the devices included in the reports. Applies only to iBeacon devices
`uidNamespaceFilter`	UID namespace Filter	A list of UID namespaces to filter devices included in the reports. Applies only Eddystone-UID devices
`urlFilter`	URL Filter	A list of URL strings to filter devices included in the reports. Applies only Eddystone-URL devices. The string listed here can be partial URL strings
`cellSizeFilter`	Cell Size Filter	A proximity filter. Devices outside the cell will not be reported. Size is specified in meters. Setting to 0 disables the cell size filter
`movementFilter`	Movement Filter	Filters devices that do not change distance. Specified in meters. Applicable only if a cell size is set. Setting to 0 disables the movement filter
`ageFilter`	Age Filter	Age filter. Devices without recent activity will not be reported
`rssiReporting`	RSSI Reporting Format	Set the preferred format for RSSI reporting

Enable the IoT Transport Profile

```
(Instant AP)(config)# iot useTransportProfile test
```

NOTE

Useful command: `#ble-init-action ble_relay send-sync-iotcfg`. This will force sync the transportProfile config with the APs.

Show Commands

To view the list of IoT transport profiles, execute the following command:

```
(host)#show iot transportProfile
```

To view the status of an IoT transport profile, execute the following command:

```
(host)#show iot transportProfile Test
```

Use the following command to view any third-party devices in the BLE table:

```
(host)#show ap debug ble-table generic
```

```
(host)#show ap debug ble-table generic | include enocean/ibeacon/sbeacon
```

To view BLE databases:

```
(host)# show ap debug ble-database
```

Viewing BLE Status

To view the BLE relay status, execute the following command (This command shows that whether connection between the Aruba AP and the server has been established, what is the Transport type, and time details):

```
(host)#show ap debug ble-relay iot-profile
```

To view the BLE relay report, execute the following command:

```
(host)# show ap debug ble-relay report <profile-name>
```

To view the BLE relay jobs, execute the following command:

```
(host)# show ap debug ble-relay jobs <profile-name>
```

To view the BLE relay status for websockets transport, execute the following command:

```
(host)# show ap debug ble-relay disp-attr all
```

To view list of BLE relay tag report, execute the following command:

```
(host)# show ap debug ble-relay tag-report <profile-name>
```

To view list of BLE relay WS log, execute the following command (This shows the detailed web socket log activity):

```
(host)# show ap debug ble-relay ws-log <profile-name>
```

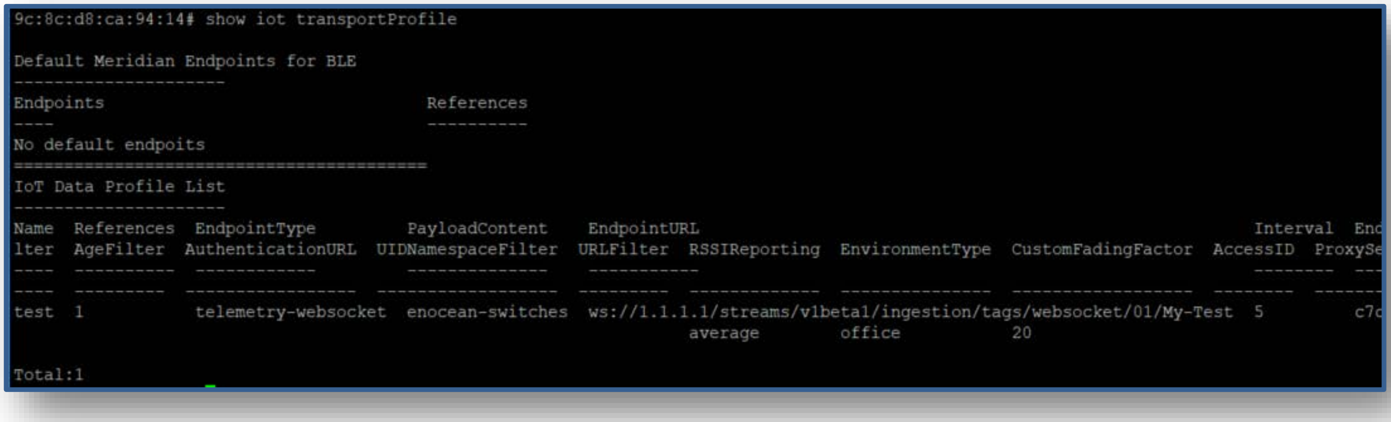
Sample Configuration and Output for Instant OS 8.6

Configuration

```
(Instant AP)# conf t
((Instant AP) (config) #iot transportProfile test
(Instant AP) (IoT Radio Profile "test") # endpointType Telemetry-Websocket
(Instant AP) (IoT Radio Profile "test") # payloadContent enocean-switches
(Instant AP) (IoT Radio Profile "test") #endpointURL
ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
(Instant AP) (IoT Radio Profile "test") # transportInterval 5
(Instant AP) (IoT Radio Profile "test")# exit

(Instant AP) (config) #iot useTransportProfile test

(Instant AP) (config)# iot radio-profile test
(Instant AP) (IoT Radio Profile "test")# radio-mode ble


(Instant AP)(config)# iot use-radio-profile test
(Instant AP)(config)# end
(Instant AP) #commit apply
```

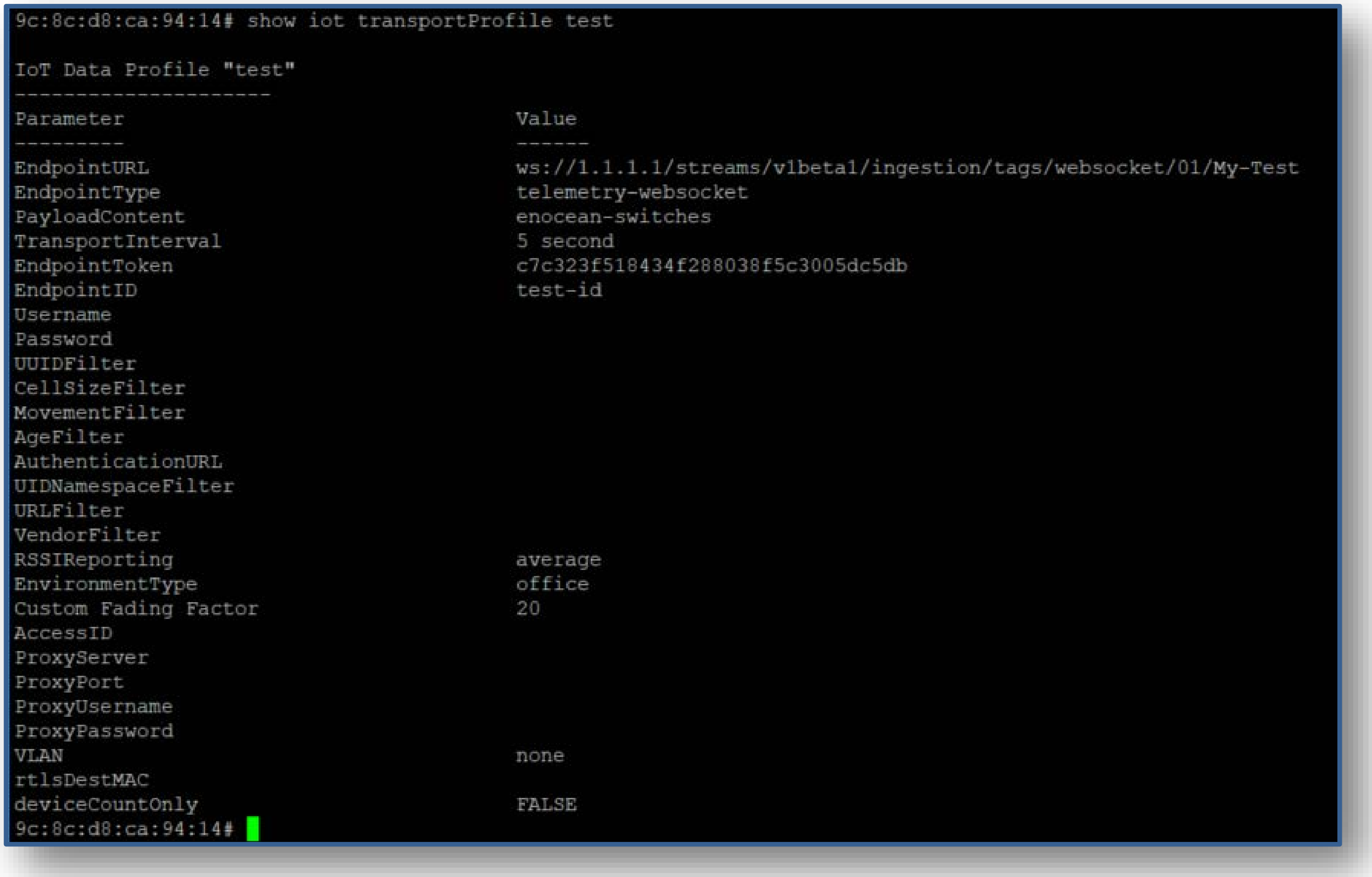
Output

#show iot transportProfile



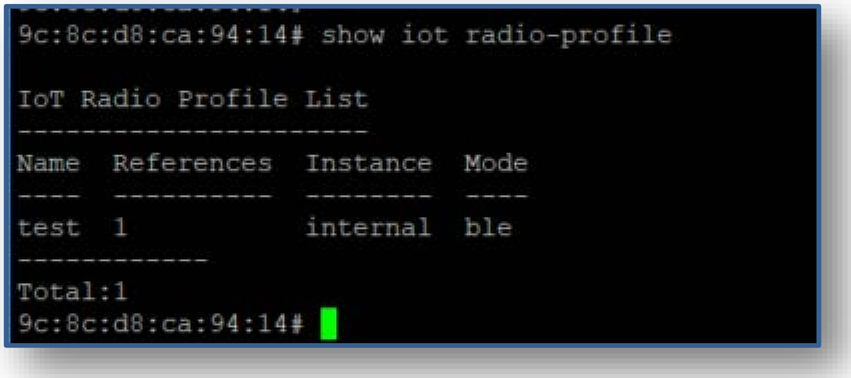
```
9c:8c:d8:ca:94:14# show iot transportProfile

Default Meridian Endpoints for BLE
---------------------
Endpoints                                References
----                                     ----------
No default endpoits
=========================================
IoT Data Profile List
---------------------
Name  References  EndpointType         PayloadContent    EndpointURL                                                       Interval  End
lter  AgeFilter  AuthenticationURL  UIDNamespaceFilter  URLFilter  RSSIReporting  EnvironmentType  CustomFadingFactor  AccessID  ProxySe
----  ----------  ------------         --------------    -----------                                                       --------  ---
----  ---------  -----------------  ------------------  ---------  -------------  ---------------  ------------------  --------  -------
test  1           telemetry-websocket  enocean-switches  ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test  5         c7c
                                                                   average        office           20

Total:1
```

# show iot transportProfile test



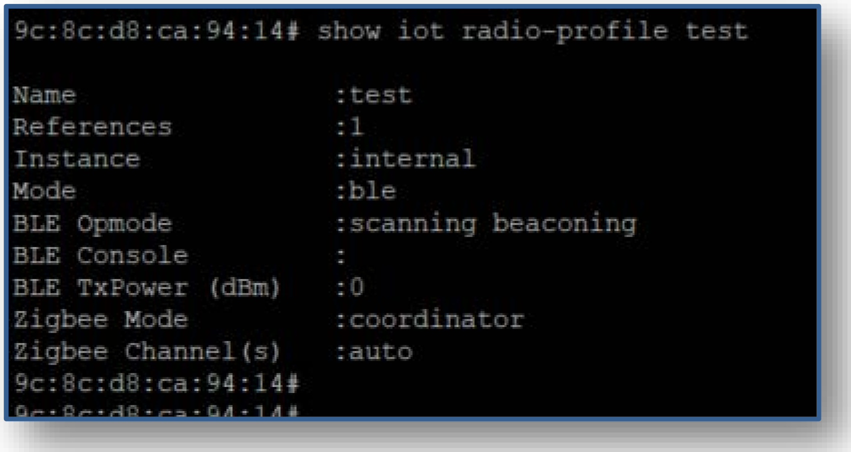
```
9c:8c:d8:ca:94:14# show iot transportProfile test

IoT Data Profile "test"
---------------------
Parameter                                Value
---------                                ------
EndpointURL                              ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
EndpointType                             telemetry-websocket
PayloadContent                           enocean-switches
TransportInterval                        5 second
EndpointToken                            c7c323f518434f288038f5c3005dc5db
EndpointID                               test-id
Username
Password
UUIDFilter
CellSizeFilter
MovementFilter
AgeFilter
AuthenticationURL
UIDNamespaceFilter
URLFilter
VendorFilter
RSSIReporting                            average
EnvironmentType                          office
Custom Fading Factor                     20
AccessID
ProxyServer
ProxyPort
ProxyUsername
ProxyPassword
VLAN                                     none
rtlsDestMAC
deviceCountOnly                          FALSE
9c:8c:d8:ca:94:14#
```

#show iot radio-profile



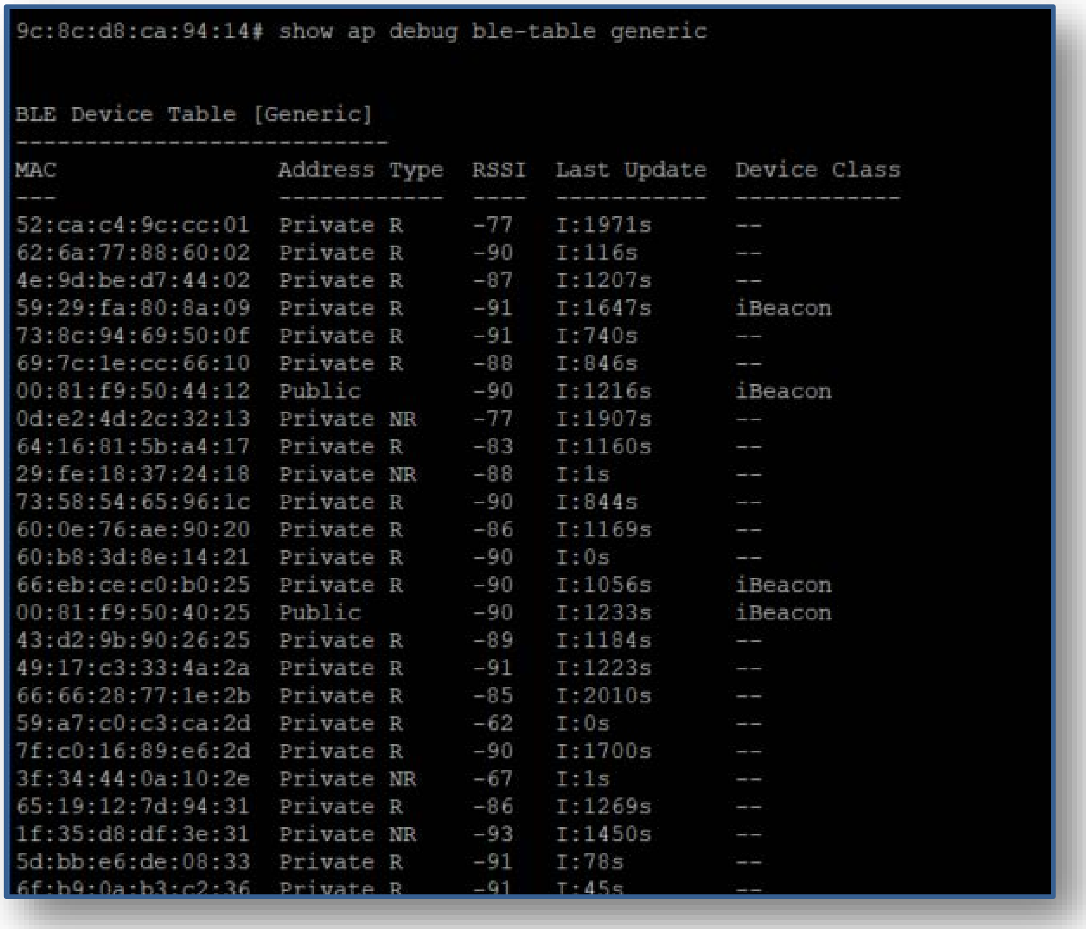
```
9c:8c:d8:ca:94:14# show iot radio-profile

IoT Radio Profile List
----------------------
Name  References  Instance  Mode
----  ----------  --------  ----
test  1           internal  ble
------------
Total:1
9c:8c:d8:ca:94:14#
```

#show iot radio-profile test



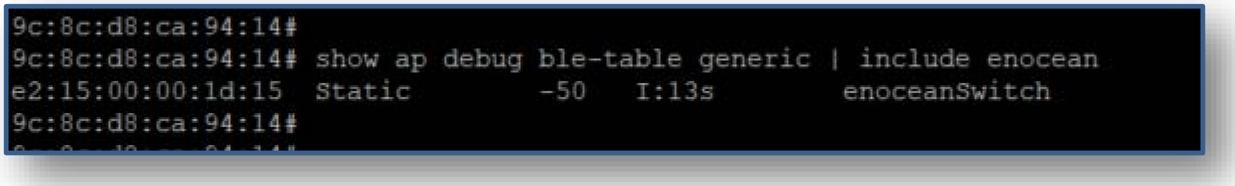
```
9c:8c:d8:ca:94:14# show iot radio-profile test

Name                    :test
References              :1
Instance                :internal
Mode                    :ble
BLE Opmode              :scanning beaconing
BLE Console             :
BLE TxPower (dBm)       :0
Zigbee Mode             :coordinator
Zigbee Channel(s)       :auto
9c:8c:d8:ca:94:14#
9c:8c:d8:ca:94:14#
```

#show ap debug ble-table generic



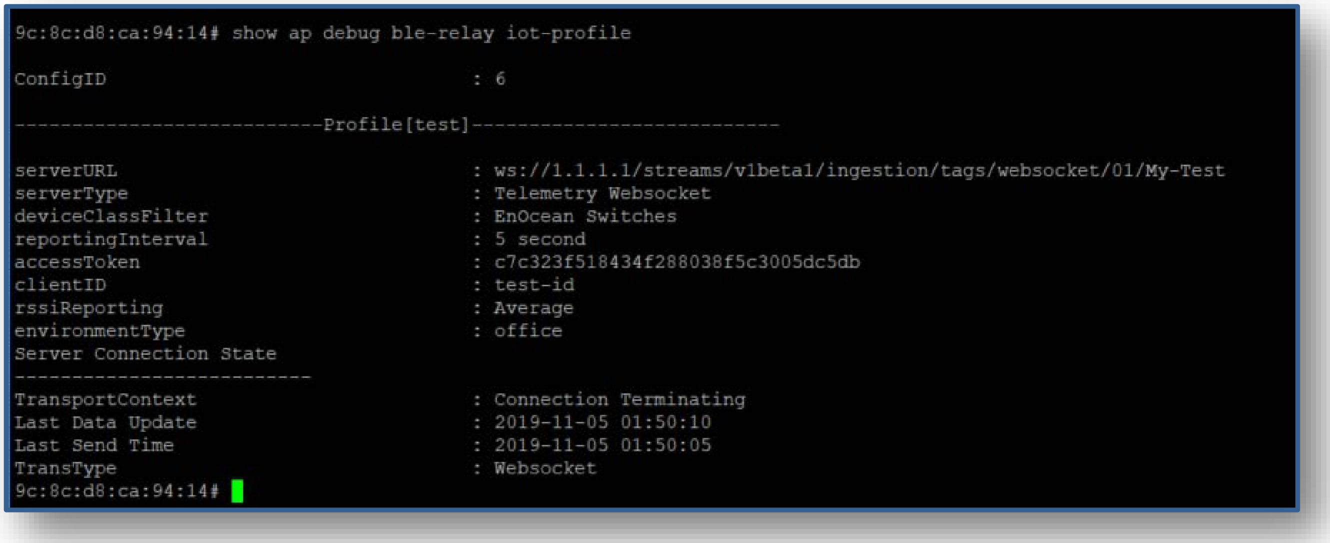
```
9c:8c:d8:ca:94:14# show ap debug ble-table generic


BLE Device Table [Generic]
---------------------------
MAC                Address Type  RSSI  Last Update  Device Class
---                ------------  ----  -----------  ------------
52:ca:c4:9c:cc:01  Private R     -77   I:1971s      --
62:6a:77:88:60:02  Private R     -90   I:116s       --
4e:9d:be:d7:44:02  Private R     -87   I:1207s      --
59:29:fa:80:8a:09  Private R     -91   I:1647s      iBeacon
73:8c:94:69:50:0f  Private R     -91   I:740s       --
69:7c:1e:cc:66:10  Private R     -88   I:846s       --
00:81:f9:50:44:12  Public        -90   I:1216s      iBeacon
0d:e2:4d:2c:32:13  Private NR    -77   I:1907s      --
64:16:81:5b:a4:17  Private R     -83   I:1160s      --
29:fe:18:37:24:18  Private NR    -88   I:1s         --
73:58:54:65:96:1c  Private R     -90   I:844s       --
60:0e:76:ae:90:20  Private R     -86   I:1169s      --
60:b8:3d:8e:14:21  Private R     -90   I:0s         --
66:eb:ce:c0:b0:25  Private R     -90   I:1056s      iBeacon
00:81:f9:50:40:25  Public        -90   I:1233s      iBeacon
43:d2:9b:90:26:25  Private R     -89   I:1184s      --
49:17:c3:33:4a:2a  Private R     -91   I:1223s      --
66:66:28:77:1e:2b  Private R     -85   I:2010s      --
59:a7:c0:c3:ca:2d  Private R     -62   I:0s         --
7f:c0:16:89:e6:2d  Private R     -90   I:1700s      --
3f:34:44:0a:10:2e  Private NR    -67   I:1s         --
65:19:12:7d:94:31  Private R     -86   I:1269s      --
1f:35:d8:df:3e:31  Private NR    -93   I:1450s      --
5d:bb:e6:de:08:33  Private R     -91   I:78s        --
6f:b9:0a:b3:c2:36  Private R     -91   I:45s        --
```

#show ap debug ble-table generic | include enocean



```
9c:8c:d8:ca:94:14#
9c:8c:d8:ca:94:14# show ap debug ble-table generic | include enocean
e2:15:00:00:1d:15  Static        -50    I:13s         enoceanSwitch
9c:8c:d8:ca:94:14#
```

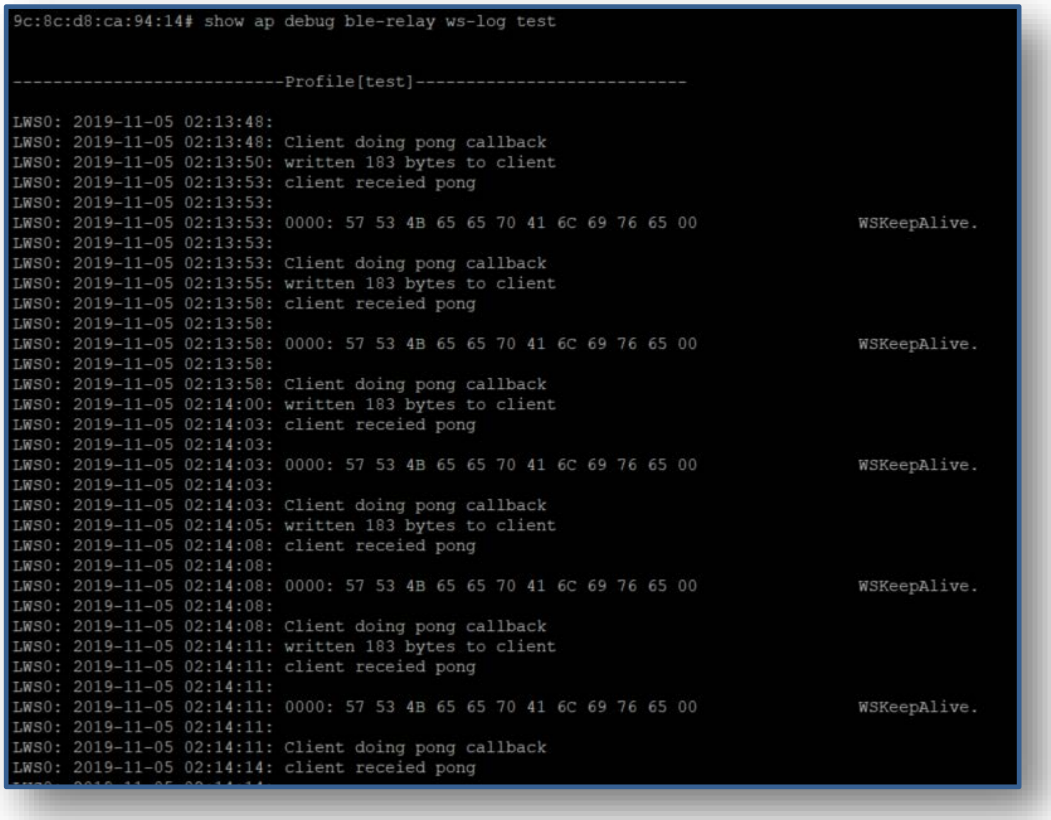
#show ap debug ble-relay iot-profile



```
9c:8c:d8:ca:94:14# show ap debug ble-relay iot-profile

ConfigID                                : 6

---------------------------Profile[test]---------------------------

serverURL                               : ws://1.1.1.1/streams/v1beta1/ingestion/tags/websocket/01/My-Test
serverType                              : Telemetry Websocket
deviceClassFilter                       : EnOcean Switches
reportingInterval                       : 5 second
accessToken                             : c7c323f518434f288038f5c3005dc5db
clientID                                : test-id
rssiReporting                           : Average
environmentType                         : office
Server Connection State
--------------------------
TransportContext                        : Connection Terminating
Last Data Update                        : 2019-11-05 01:50:10
Last Send Time                          : 2019-11-05 01:50:05
TransType                               : Websocket
9c:8c:d8:ca:94:14#
```

*Once the connection is established, you should see “**TransportContext: Connection Established**”. This is the indicator that successful communication is happening between Aruba controller and the Web socket server.

#show ap debug ble-relay ws-log test



```
9c:8c:d8:ca:94:14# show ap debug ble-relay ws-log test


---------------------------Profile[test]---------------------------

LWS0: 2019-11-05 02:13:48:
LWS0: 2019-11-05 02:13:48: Client doing pong callback
LWS0: 2019-11-05 02:13:50: written 183 bytes to client
LWS0: 2019-11-05 02:13:53: client receied pong
LWS0: 2019-11-05 02:13:53:
LWS0: 2019-11-05 02:13:53: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-05 02:13:53:
LWS0: 2019-11-05 02:13:53: Client doing pong callback
LWS0: 2019-11-05 02:13:55: written 183 bytes to client
LWS0: 2019-11-05 02:13:58: client receied pong
LWS0: 2019-11-05 02:13:58:
LWS0: 2019-11-05 02:13:58: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-05 02:13:58:
LWS0: 2019-11-05 02:13:58: Client doing pong callback
LWS0: 2019-11-05 02:14:00: written 183 bytes to client
LWS0: 2019-11-05 02:14:03: client receied pong
LWS0: 2019-11-05 02:14:03:
LWS0: 2019-11-05 02:14:03: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-05 02:14:03:
LWS0: 2019-11-05 02:14:03: Client doing pong callback
LWS0: 2019-11-05 02:14:05: written 183 bytes to client
LWS0: 2019-11-05 02:14:08: client receied pong
LWS0: 2019-11-05 02:14:08:
LWS0: 2019-11-05 02:14:08: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-05 02:14:08:
LWS0: 2019-11-05 02:14:08: Client doing pong callback
LWS0: 2019-11-05 02:14:11: written 183 bytes to client
LWS0: 2019-11-05 02:14:11: client receied pong
LWS0: 2019-11-05 02:14:11:
LWS0: 2019-11-05 02:14:11: 0000: 57 53 4B 65 65 70 41 6C 69 76 65 00                WSKeepAlive.
LWS0: 2019-11-05 02:14:11:
LWS0: 2019-11-05 02:14:11: Client doing pong callback
LWS0: 2019-11-05 02:14:14: client receied pong
```

Zigbee

This section covers IoT configuration for Zigbee on AOS and Instant. It covers the steps for enabling Zigbee radio, radio profile and transport profile. It also covers some helpful monitoring commands and finally goes through sample configurations and show command output.

AOS

Enable Zigbee

For AOS versions 8.6 and above

```
(host) [mynode] (config)# iot radio-profile aa-radio-profile
(host) [mynode] (IoT Radio Profile "aa-radio-profile")# radio-mode zigbee
```

Radio Profile Configuration Options

Config Option	Description
ble-console	BLE console mode
ble-opmode	BLE operational mode
ble-txpower	BLE transmit power in dBm
clone	Copy data from another IoT Radio Profile
no	Delete Command
radio-instance	IoT radio instance
radio-mode	IoT radio modes
zigbee-channel	ZigBee scanning channel[auto, 11-26]
zigbee-opmode	ZigBee operational mode

Configure Zigbee Service Profile

```
(host) [mynode] (config)# zigbee service-profile aa-service-profile
(host) [mynode] (ZigBee Service Profile "aa-service-profile")# radio-instance in-
ternal
(host) [mynode] (ZigBee Service Profile "aa-service-profile")# security disable
```

Enable the IoT Radio Profile and Zigbee Service Profile (For AOS 8.6 and above)

```
(host) [mynode] (ZigBee Service Profile "aa-service-profile")#ap-group <default>
(host) [mynode](AP group "default") #iot radio-profile aa-radio-profile
(host) [mynode](AP group "default") #zigbee service-profile aa-service-profile
```

Configure IoT Transport Profile

Required Account Details

Following values are needed from your server before configuring the Transport Profile:

- serverURL
- username
- clientID
- password

Mandatory Settings

Following parameters are required to be configure in your Transport Profile mandatorily (Example shown in the next section):

- serverType
- clientID
- serverURL
- username/password
- include-ap-group

Recommended Settings

Following parameters can be configured for your Transport Profile optionally (Example shown in the next section):

- deviceClass filter
- reportingPeriod

Remaining fields can be left as default.

Configuration:

```
(host) [mynode] (config) #iot transportProfile aa
(host) [mynode] (IoT Data Profile "aa") #serverType Assa-Abloy
(host) [mynode] (IoT Data Profile "aa") #serverURL <URL>
```

Sample URL: https://10.10.10.10/443

```
(host) [mynode] (IoT Data Profile "aa") #username <username for your server>
(host) [mynode] (IoT Data Profile "aa") #password <password for your server>
(host) [mynode] (IoT Data Profile "aa") #clientID <clientID for your server>
```

Sample clientID: 1234567890

```
(host) [mynode] (IoT Data Profile "aa") #include-ap-group <default>

(host) [mynode] (IoT Data Profile "aa") #deviceClassFilter assa-abloy

(host) [mynode] (IoT Data Profile "aa") #reportingInterval 10
```

NOTE Username, password, clientID and serverURL will be the credentials of your server.

Transport Profile Configuration Options:

Config Option	Short Description	Long Description
`serverType`	Server Type	The type of server that is receiving the telemetry stream
`authenticationURL`	Authentication URL	Server URL for authentication
`username`	Username	Username for authentication
`password`	Password	Password for authentication
`clientID`	Client ID	This ID identifies the sender to the server
`serverURL`	Server URL	Server URL for sending telemetry
`accessToken`	Access Token	Access token. Configure this only if you want to bypass authentication
`reportingInterval`	Reporting Interval	Reporting interval in seconds
`deviceClassFilter`	Device Class Filter	A list of device class tags to filter the devices included in the reports
`uuidFilter`	UUID Filter	A list of UUIDs to filter the devices included in the reports. Applies only to iBeacon devices
`uidNamespaceFilter`	UID namespace Filter	A list of UID namespaces to filter devices included in the reports. Applies only Eddystone-UID devices
`urlFilter`	URL Filter	A list of URL strings to filter devices included in the reports. Applies only Eddystone-URL devices. The string listed here can be partial URL strings
`cellSizeFilter`	Cell Size Filter	A proximity filter. Devices outside the cell will not be reported. Size is specified in meters. Setting to 0 disables the cell size filter
`movementFilter`	Movement Filter	Filters devices that do not change distance. Specified in meters. Applicable only if a cell size is set. Setting to 0 disables the movement filter
`ageFilter`	Age Filter	Age filter. Devices without recent activity will not be reported
`rssiReporting`	RSSI Reporting Format	Set the preferred format for RSSI reporting

Enable the IoT Transport Profile

To apply an IoT transport profile, execute the following command:

```
(host) [mynode] (config) #iot useTransportProfile aa
(host) [mynode] (config) #exit
(host) [mynode] # write memory
```

Enable Zigbee permit-joining

```
(host) [mynode] (config) #ap zigbee-init-action permit-joining ap-name <ap-name>
radio all restart <duration 60-600 seconds>
```

NOTE

Enabling permit-joining options opens a window of 600 seconds by default (a timer between 60 to 600 seconds can be set) where a new zigbee device can join APs zigbee radio for communication. After enabling this, we can associate our Discovery card with the door lock to the AP. We will be able to see the door lock as a zigbee client after this action.

NOTE

Once the intended zigbee devices are associated, following command can be used optionally to stop the permit-joining window instantly:

```
(host) [mynode] (config) #ap zigbee-init-action permit-joining ap-name <ap-name>
radio all stop.
```

Show Commands

To view the list of IoT transport profiles, execute the following command:

```
(host) [mynode] (config) #show iot transportProfile
```

To view the status of an IoT transport profile, execute the following command:

```
(host) [mynode] (config) #show iot transportProfile aa
```

To view AP name, execute the following command:

```
(host) [mynode] (config) #show ap database
```

To view the list of IoT radio profiles, execute the following command:

```
(host) [mynode] (config) #show iot radio-profile
```

To view the status of an IoT radio profile, execute the following command:

```
(host) [mynode] (config) #show iot radio-profile aa-radio-profile
```

To view the status of Zigbee service profile, execute the following command:

```
(host) [mynode] (config) #show zigbee service-profile aa-service-profile
```

To view what radio and service profiles are applied to an AP-Group:

```
(host) [mynode] (config) #show ap-group default
```

Viewing Zigbee Status

Use the following command to view Zigbee's client table:

```
(host) [mynode] (config) # show ap debug zigbee ap-name AP1 client-table
```

Use the following command to view Zigbee's radio table. This shows AP's own zigbee radio:

```
(host) [mynode] (config)# show ap debug zigbee ap-name AP1 radio-table
```

Use the following command to view Zigbee's socket table info:

```
(host) [mynode] (config)# show ap debug zigbee ap-name AP1 socket-table
```

Use the following command to view Zigbee's packet trail info:

```
(host) [mynode] (config)# show ap debug zigbee ap-name AP1 packet-trail
```

To view the Zigbee relay status, execute the following command (This command shows that whether connection between the Aruba AP and the server has been established, what is the Transport type, and time details):

```
(host) [mynode] #show ble_relay iot-profile
```

To view the Zigbee relay report, execute the following command:

```
(host) [mynode] #show ble_relay report <transport-profile-name>
```

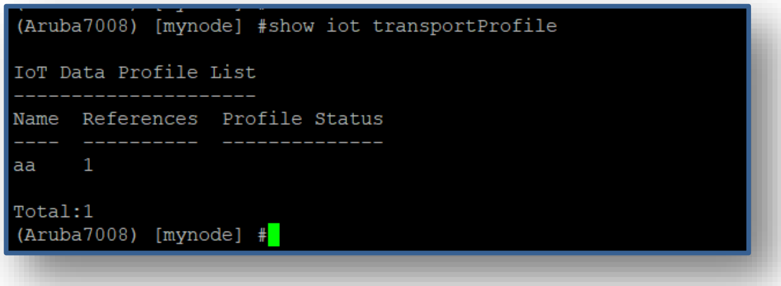
Sample Configuration and Output for AOS 8.7

Configuration

```
(host) [mynode] (config)# iot radio-profile aa-radio-profile
(host) [mynode] (IoT Radio Profile "aa-radio-profile")# radio-mode zigbee
(host) [mynode] (IoT Radio Profile "aa-radio-profile")# exit

(host) [mynode] (config)# zigbee service-profile aa-service-profile
(host) [mynode] (ZigBee Service Profile "aa-service-profile")# radio-instance in-
ternal
(host) [mynode] (ZigBee Service Profile "aa-service-profile")# security disable


(host) [mynode] (ZigBee Service Profile "aa-service-profile")#ap-group default
(host) [mynode](AP group "default") #iot radio-profile aa-radio-profile
(host) [mynode](AP group "default") #zigbee service-profile aa-service-profile
(host) [mynode](AP group "default") #exit


(host) [mynode] (config) #iot transportProfile aa
(host) [mynode] (IoT Data Profile "aa") #serverType Assa-Abloy
(host) [mynode] (IoT Data Profile "aa") #serverURL https://10.10.10.10/443
(host) [mynode] (IoT Data Profile "aa") #username admin
(host) [mynode] (IoT Data Profile "aa") #password Admin123
(host) [mynode] (IoT Data Profile "aa") #clientID 1234567890
(host) [mynode] (IoT Data Profile "aa") #include-ap-group default
(host) [mynode] (IoT Data Profile "aa") #deviceClassFilter assa-abloy
(host) [mynode] (IoT Data Profile "aa") #reportingInterval 10
(host) [mynode] (IoT Data Profile "aa") #exit

(host) [mynode] (config) #iot useTransportProfile aa
(host) [mynode] (config) # write memory


(host) [mynode] (config) #ap zigbee-init-action permit-joining ap-name AP515 radio
all restart 600
```

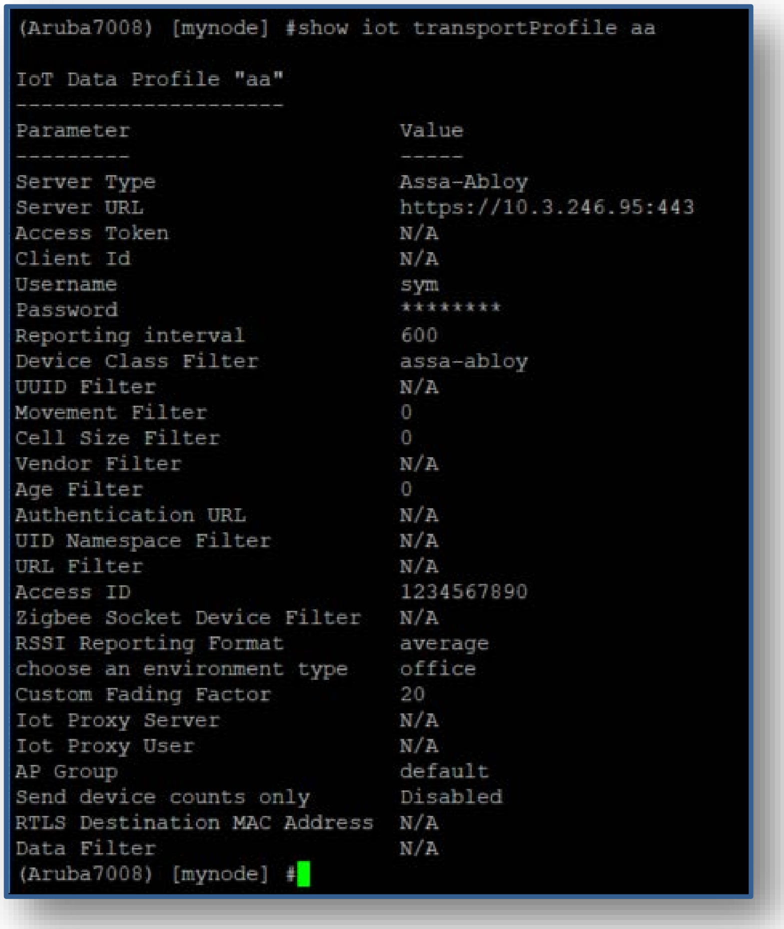
Output

#show iot transportProfile



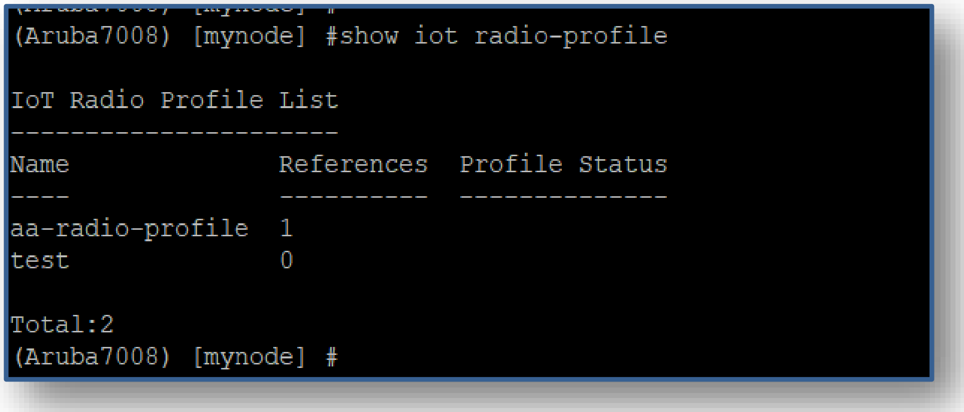
```
(Aruba7008) [mynode] #show iot transportProfile

IoT Data Profile List
---------------------
Name  References  Profile Status
----  ----------  --------------
aa    1

Total:1
(Aruba7008) [mynode] #
```

show iot transportProfile aa



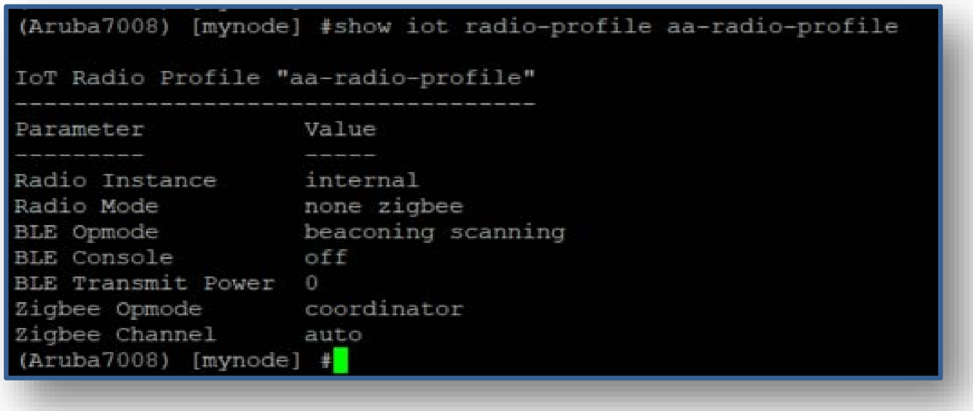
```
(Aruba7008) [mynode] #show iot transportProfile aa

IoT Data Profile "aa"
---------------------
Parameter                     Value
---------                     -----
Server Type                   Assa-Abloy
Server URL                    https://10.3.246.95:443
Access Token                  N/A
Client Id                     N/A
Username                      sym
Password                      ********
Reporting interval            600
Device Class Filter           assa-abloy
UUID Filter                   N/A
Movement Filter               0
Cell Size Filter              0
Vendor Filter                 N/A
Age Filter                    0
Authentication URL            N/A
UID Namespace Filter          N/A
URL Filter                    N/A
Access ID                     1234567890
Zigbee Socket Device Filter   N/A
RSSI Reporting Format         average
choose an environment type    office
Custom Fading Factor          20
Iot Proxy Server              N/A
Iot Proxy User                N/A
AP Group                      default
Send device counts only       Disabled
RTLS Destination MAC Address  N/A
Data Filter                   N/A
(Aruba7008) [mynode] #
```

#show iot radio-profile



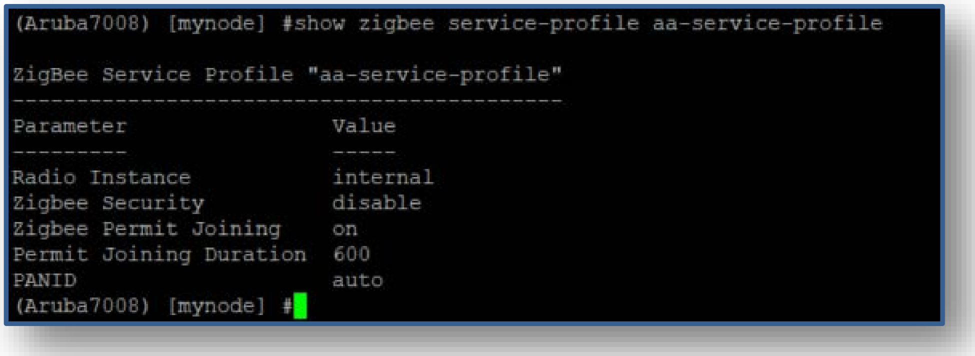
```
(Aruba7008) [mynode] #
(Aruba7008) [mynode] #show iot radio-profile

IoT Radio Profile List
----------------------
Name               References  Profile Status
----               ----------  --------------
aa-radio-profile   1
test               0

Total:2
(Aruba7008) [mynode] #
```

#show iot radio-profile aa-radio-profile



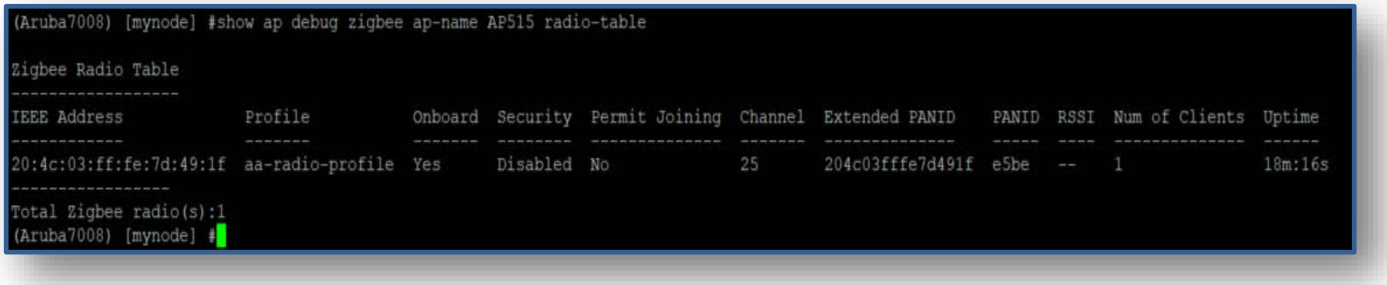
```
(Aruba7008) [mynode] #show iot radio-profile aa-radio-profile

IoT Radio Profile "aa-radio-profile"
-----------------------------------
Parameter            Value
---------            -----
Radio Instance       internal
Radio Mode           none zigbee
BLE Opmode           beaconing scanning
BLE Console          off
BLE Transmit Power   0
Zigbee Opmode        coordinator
Zigbee Channel       auto
(Aruba7008) [mynode] #
```

show zigbee service-profile aa-service-profile



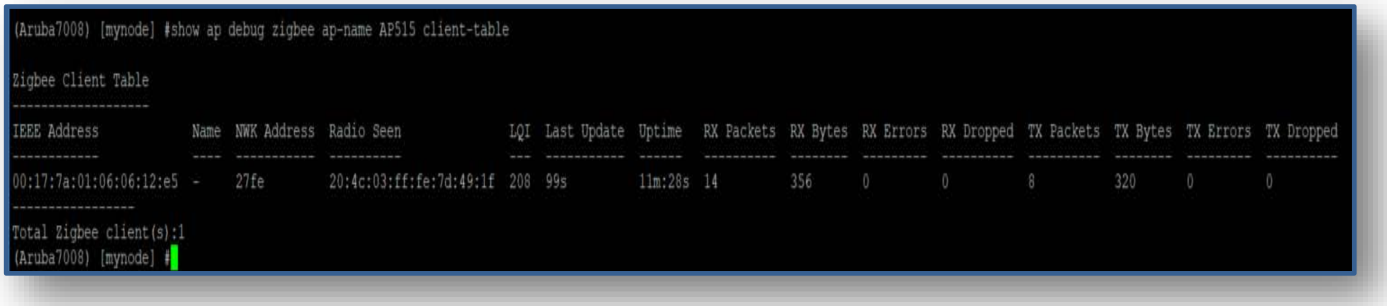
```
(Aruba7008) [mynode] #show zigbee service-profile aa-service-profile

ZigBee Service Profile "aa-service-profile"
-------------------------------------------
Parameter                Value
---------                -----
Radio Instance           internal
Zigbee Security          disable
Zigbee Permit Joining    on
Permit Joining Duration  600
PANID                    auto
(Aruba7008) [mynode] #
```

show ap debug zigbee ap-name AP515 radio-table



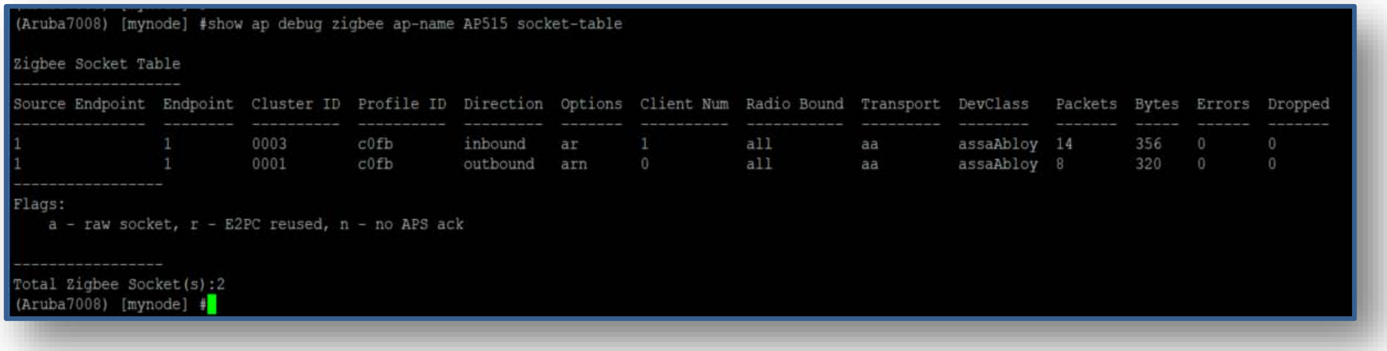
```
(Aruba7008) [mynode] #show ap debug zigbee ap-name AP515 radio-table

Zigbee Radio Table
------------------
IEEE Address            Profile           Onboard  Security  Permit Joining  Channel  Extended PANID    PANID  RSSI  Num of Clients  Uptime
------------            -------           -------  --------  --------------  -------  --------------    -----  ----  --------------  ------
20:4c:03:ff:fe:7d:49:1f  aa-radio-profile  Yes      Disabled  No              25       204c03fffe7d491f  e5be   --    1               18m:16s
-----------------
Total Zigbee radio(s):1
(Aruba7008) [mynode] #
```

show ap debug zigbee ap-name AP515 client-table



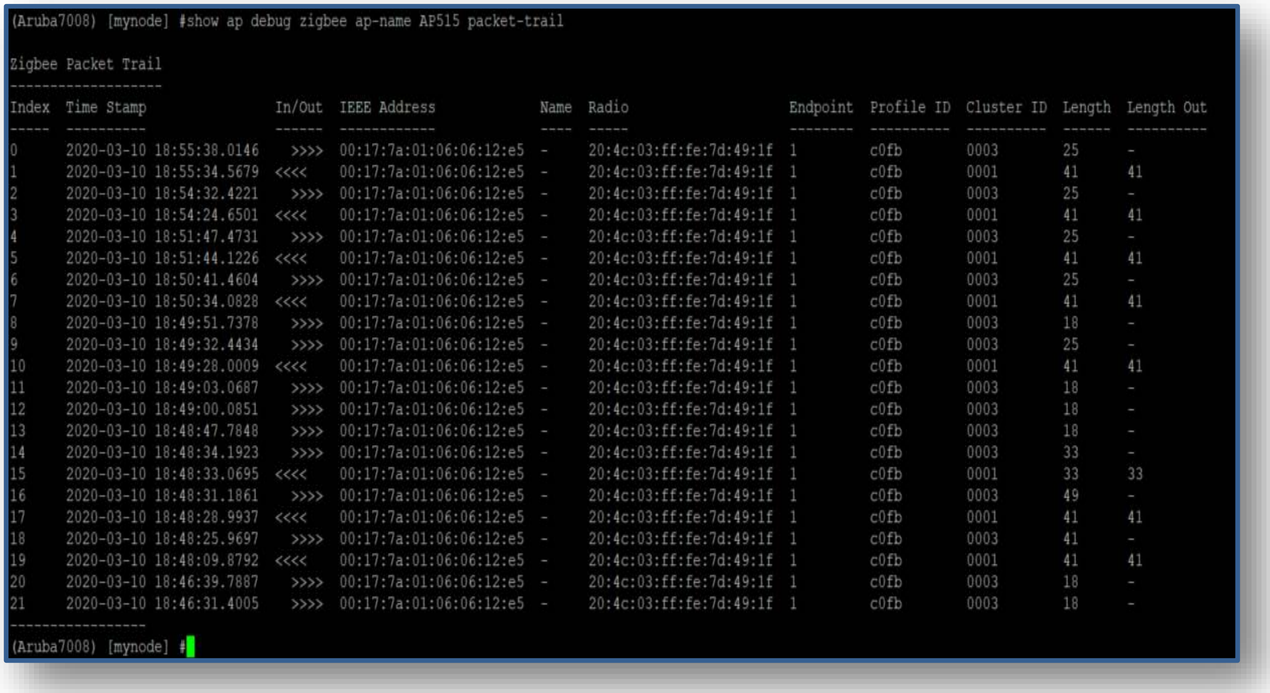
```
(Aruba7008) [mynode] #show ap debug zigbee ap-name AP515 client-table

Zigbee Client Table
-------------------
IEEE Address             Name  NWK Address  Radio Seen               LQI  Last Update  Uptime   RX Packets  RX Bytes  RX Errors  RX Dropped  TX Packets  TX Bytes  TX Errors  TX Dropped
------------             ----  -----------  ----------               ---  -----------  ------   ----------  --------  ---------  ----------  ----------  --------  ---------  ----------
00:17:7a:01:06:06:12:e5  -     27fe         20:4c:03:ff:fe:7d:49:1f  208  99s          11m:28s  14          356       0          0           8           320       0          0
-----------------
Total Zigbee client(s):1
(Aruba7008) [mynode] #
```

show ap debug zigbee ap-name AP515 socket-table



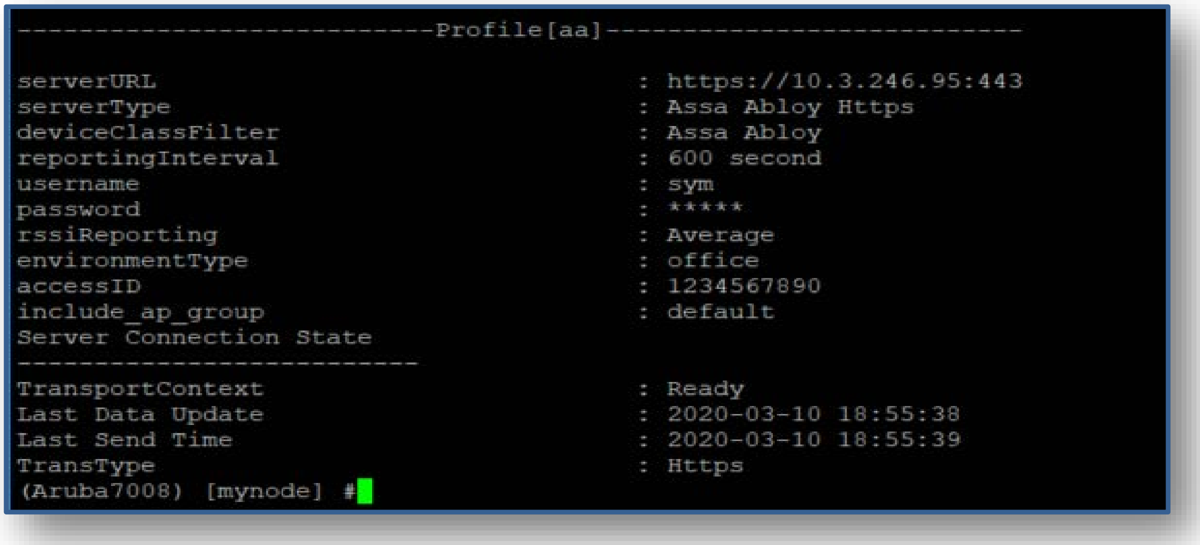
```
(Aruba7008) [mynode] #show ap debug zigbee ap-name AP515 socket-table

Zigbee Socket Table
-------------------
Source Endpoint  Endpoint  Cluster ID  Profile ID  Direction  Options  Client Num  Radio Bound  Transport  DevClass   Packets  Bytes  Errors  Dropped
---------------  --------  ----------  ----------  ---------  -------  ----------  -----------  ---------  --------   -------  -----  ------  -------
1                1         0003        c0fb        inbound    ar       1           all          aa         assaAbloy  14       356    0       0
1                1         0001        c0fb        outbound   arn      0           all          aa         assaAbloy  8        320    0       0
-----------------
Flags:
    a - raw socket, r - E2PC reused, n - no APS ack

-----------------
Total Zigbee Socket(s):2
(Aruba7008) [mynode] #
```

show ap debug zigbee ap-name AP515 packet-trail



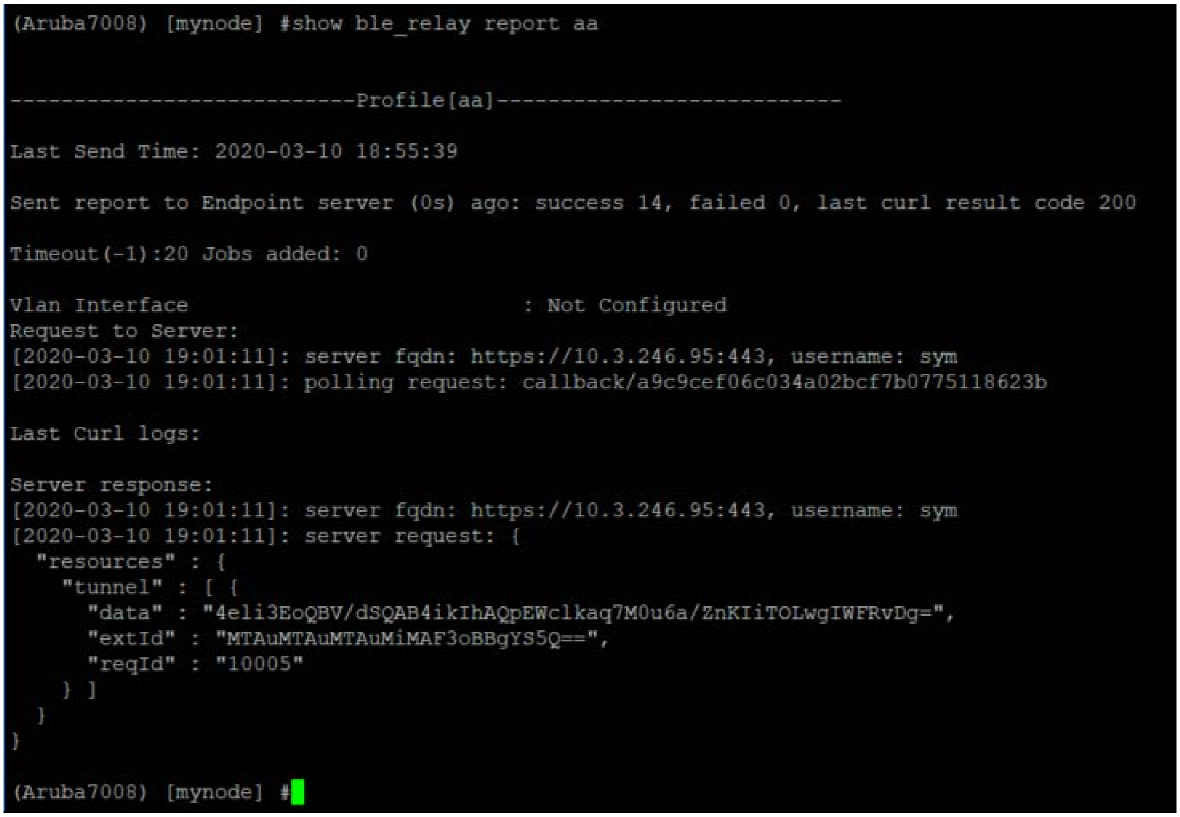
```
(Aruba7008) [mynode] #show ap debug zigbee ap-name AP515 packet-trail

Zigbee Packet Trail
-------------------
Index  Time Stamp               In/Out  IEEE Address             Name  Radio                    Endpoint  Profile ID  Cluster ID  Length  Length Out
-----  ----------               ------  ------------             ----  -----                    --------  ----------  ----------  ------  ----------
0      2020-03-10 18:55:38.0146   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        25      -
1      2020-03-10 18:55:34.5679 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
2      2020-03-10 18:54:32.4221   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        25      -
3      2020-03-10 18:54:24.6501 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
4      2020-03-10 18:51:47.4731   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        25      -
5      2020-03-10 18:51:44.1226 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
6      2020-03-10 18:50:41.4604   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        25      -
7      2020-03-10 18:50:34.0828 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
8      2020-03-10 18:49:51.7378   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
9      2020-03-10 18:49:32.4434   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        25      -
10     2020-03-10 18:49:28.0009 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
11     2020-03-10 18:49:03.0687   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
12     2020-03-10 18:49:00.0851   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
13     2020-03-10 18:48:47.7848   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
14     2020-03-10 18:48:34.1923   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        33      -
15     2020-03-10 18:48:33.0695 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        33      33
16     2020-03-10 18:48:31.1861   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        49      -
17     2020-03-10 18:48:28.9937 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
18     2020-03-10 18:48:25.9697   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        41      -
19     2020-03-10 18:48:09.8792 <<<<    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0001        41      41
20     2020-03-10 18:46:39.7887   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
21     2020-03-10 18:46:31.4005   >>>>  00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:49:1f  1         c0fb        0003        18      -
-----------------
(Aruba7008) [mynode] #
```

#show ble_relay iot-profile



```
---------------------------Profile[aa]---------------------------

serverURL                               : https://10.3.246.95:443
serverType                              : Assa Abloy Https
deviceClassFilter                       : Assa Abloy
reportingInterval                       : 600 second
username                                : sym
password                                : *****
rssiReporting                           : Average
environmentType                         : office
accessID                                : 1234567890
include_ap_group                        : default
Server Connection State
--------------------------
TransportContext                        : Ready
Last Data Update                        : 2020-03-10 18:55:38
Last Send Time                          : 2020-03-10 18:55:39
TransType                               : Https
(Aruba7008) [mynode] #
```

#show ble_relay report aa



```
(Aruba7008) [mynode] #show ble_relay report aa


---------------------------Profile[aa]---------------------------

Last Send Time: 2020-03-10 18:55:39

Sent report to Endpoint server (0s) ago: success 14, failed 0, last curl result code 200

Timeout(-1):20 Jobs added: 0

Vlan Interface                          : Not Configured
Request to Server:
[2020-03-10 19:01:11]: server fqdn: https://10.3.246.95:443, username: sym
[2020-03-10 19:01:11]: polling request: callback/a9c9cef06c034a02bcf7b0775118623b

Last Curl logs:

Server response:
[2020-03-10 19:01:11]: server fqdn: https://10.3.246.95:443, username: sym
[2020-03-10 19:01:11]: server request: {
  "resources" : {
    "tunnel" : [ {
      "data" : "4eli3EoQBV/dSQAB4ikIhAQpEWclkaq7M0u6a/ZnKIiTOLwgIWFRvDg=",
      "extId" : "MTAuMTAuMTAuMiMAF3oBBgYS5Q==",
      "reqId" : "10005"
    } ]
  }
}

(Aruba7008) [mynode] #
```

Instant

Enable Zigbee

For Instant OS versions 8.6 and above

```
(Instant AP) # configure terminal
(Instant AP)(config)# iot radio-profile aa-radio-profile
(Instant AP) (IoT Radio Profile "aa-radio-profile") #radio-mode zigbee
(Instant AP) (IoT Radio Profile "aa-radio-profile") #exit
```

Enable the IoT Radio Profile (For Instant OS 8.6 and above)

```
(Instant AP)(config)# iot use-radio-profile aa-radio-profile
```

Configure Zigbee Service Profile

```
(Instant AP)(config)# zigbee service-profile aa-service-profile
(Instant AP)(Zigbee Service Profile "aa-service-profile")# radio-instance internal
(Instant AP)(Zigbee Service Profile "aa-service-profile")# security disable
```

Enable the Zigbee Service Profile (For Instant OS 8.6 and above)

```
(Instant AP)(config)# zigbee use-service-profile aa-service-profile
```

Configure IoT Transport Profile

Required Account Details

Following values are needed from your server before configuring the Transport Profile:

- endpointURL
- password
- endpointID
- username

Mandatory Settings

Following parameters are required to be configure in your Transport Profile mandatorily:

- endpointType
- endpointID
- endpointURL
- username/password

Recommended Settings

Following parameters can be configured for your Transport Profile optionally (Example shown in the next section):

- payloadContent
- transportInterval

Remaining fields can be left as default.

Configuration

```
(Instant AP)(config)# iot transportProfile aa
(Instant AP)(IoT Transport Profile "aa")# endpointType Assa-Abloy
(Instant AP)(IoT Transport Profile "aa")# endPointURL <URL>
```

Sample URL: https://10.10.10.10/443

```
(Instant AP)(IoT Transport Profile "aa")# username <username for your server>
(Instant AP)(IoT Transport Profile "aa")# password <password for your server>
(Instant AP)(IoT Transport Profile "aa")# endpointID <clientID for your server>
```

Sample clientID: 1234567890

```
(Instant AP)(IoT Transport Profile "aa")# payloadContent assa-abloy
(Instant AP)(IoT Transport Profile "aa")# transportInterval 10
(Instant AP)(IoT Transport Profile "aa")# exit
```

Username, password, endpointID and endpointURL will be the credentials of your server.

Transport Profile Configuration Options

Config Option	Short Description	Long Description
`endpointType`	Server Type	The type of server that is receiving the telemetry stream
`authenticationURL`	Authentication URL	Server URL for authentication
`username`	Username	Username for authentication
`password`	Password	Password for authentication
`endpointID`	Client ID	This ID identifies the sender to the server
`endpointURL`	Server URL	Server URL for sending telemetry
`endpointToken`	Access Token	Access token. Configure this only if you want to by-pass authentication

`transportInterval`	Reporting Interval	Reporting interval in seconds
`payloadContent`	Device Class Filter	A list of device class tags to filter the devices included in the reports
`uuidFilter`	UUID Filter	A list of UUIDs to filter the devices included in the reports. Applies only to iBeacon devices
`uidNamespaceFilter`	UID namespace Filter	A list of UID namespaces to filter devices included in the reports. Applies only Eddystone-UID devices
`urlFilter`	URL Filter	A list of URL strings to filter devices included in the reports. Applies only Eddystone-URL devices. The string listed here can be partial URL strings
`cellSizeFilter`	Cell Size Filter	A proximity filter. Devices outside the cell will not be reported. Size is specified in meters. Setting to 0 disables the cell size filter
`movementFilter`	Movement Filter	Filters devices that do not change distance. Specified in meters. Applicable only if a cell size is set. Setting to 0 disables the movement filter
`ageFilter`	Age Filter	Age filter. Devices without recent activity will not be reported
`rssiReporting`	RSSI Reporting Format	Set the preferred format for RSSI reporting

Enable the IoT Transport Profile

```
(Instant AP)(config)# iot useTransportProfile aa
```

```
(Instant AP)#commit apply
```

Enable Zigbee permit-joining

```
(Instant AP)# zigbee-init-action permit-joining radio all restart <duration 60-600
seconds>
```

Enabling permit-joining options opens a window of 300 seconds by default (a timer between 60 to 600 seconds can be set) where a new zigbee device can join APs zigbee radio for communication. After enabling this, we can associate our Discovery card with the door lock to the AP. We will be able to see the door lock as a zigbee client after this action.

Once the intended zigbee devices are associated, following command can be used optionally to stop the permit-joining window instantly:

```
(Instant AP)# zigbee-init-action permit-joining radio all stop.
```

Show Commands

To view the list of IoT transport profiles, execute the following command:

```
(host)#show iot transportProfile
```

To view the status of an IoT transport profile, execute the following command:

```
(host)#show iot transportProfile aa
```

To view the list of IoT radio profiles, execute the following command:

```
(host)#show iot radio-profile
```

To view the status of an IoT radio profile, execute the following command:

```
(host)#show iot radio-profile aa-radio-profile
```

To view the list of zigbee service-profiles, execute the following command:

```
(host)#show zigbee service-profile
```

To view the status of Zigbee service profile, execute the following command:

```
(host)#show zigbee service-profile aa-service-profile
```

Viewing Zigbee Status

To view the BLE relay status, execute the following command (This command shows that whether connection between the Aruba AP and the server has been established, what is the Transport type, and time details):

```
(host)#show ap debug ble-relay iot-profile
```

To view the BLE relay report, execute the following command:

```
(host)# show ap debug ble-relay report <transport-profile-name>
```

Use the following command to view Zigbee's client table:

```
(host)# show ap debug zigbee client-table
```

Use the following command to view Zigbee's radio table. This shows AP's own zigbee radio:

```
(host)# show ap debug zigbee radio-table
```

Use the following command to view Zgbee's socket table info:

```
(host)# show ap debug zigbee socket-table
```

Use the following command to view Zigbee's packet trail info:

```
(host)# show ap debug zigbee packet-trail
```

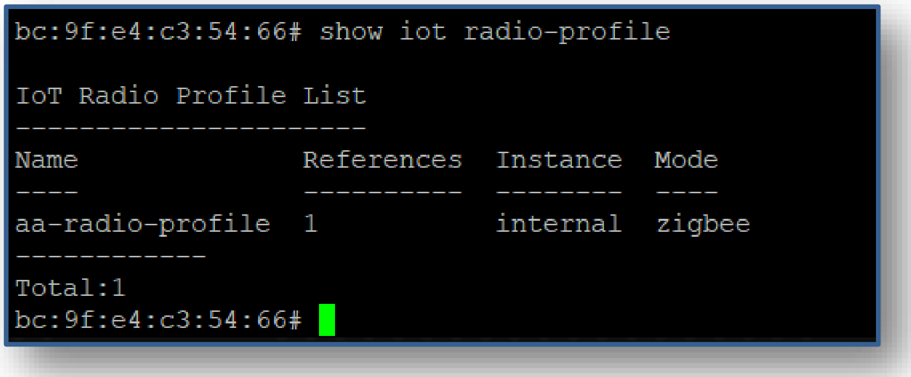
Sample Configuration and Output for Instant OS 8.7

Configuration:

```
(Instant AP) # configure terminal
(Instant AP)(config)# iot radio-profile aa-radio-profile
(Instant AP) (IoT Radio Profile "aa-radio-profile") #radio-mode zigbee
(Instant AP) (IoT Radio Profile "aa-radio-profile") #exit

(Instant AP)(config)# iot use-radio-profile aa-radio-profile

(Instant AP)(config)# zigbee service-profile aa-service-profile
(Instant AP)(Zigbee Service Profile "aa-service-profile")# radio-instance internal
(Instant AP)(Zigbee Service Profile "aa-service-profile")# security disable
(Instant AP) (IoT Radio Profile "aa-service-profile") #exit

(Instant AP)(config)# iot transportProfile aa
(Instant AP)(IoT Transport Profile "aa")# endpointType Assa-Abloy
(Instant AP)(IoT Transport Profile "aa")# endPointURL https://10.10.10.10/443
(Instant AP)(IoT Transport Profile "aa")# username admin
(Instant AP)(IoT Transport Profile "aa")# password Admin123
(Instant AP)(IoT Transport Profile "aa")# endpointID 1234567890

(Instant AP)(IoT Transport Profile "aa")# payloadContent assa-abloy
(Instant AP)(IoT Transport Profile "aa")# transportInterval 10
(Instant AP)(IoT Transport Profile "aa")# exit

(Instant AP)(config)# iot useTransportProfile aa
(Instant AP)#commit apply

(Instant AP)# zigbee-init-action permit-joining radio all restart 600
```

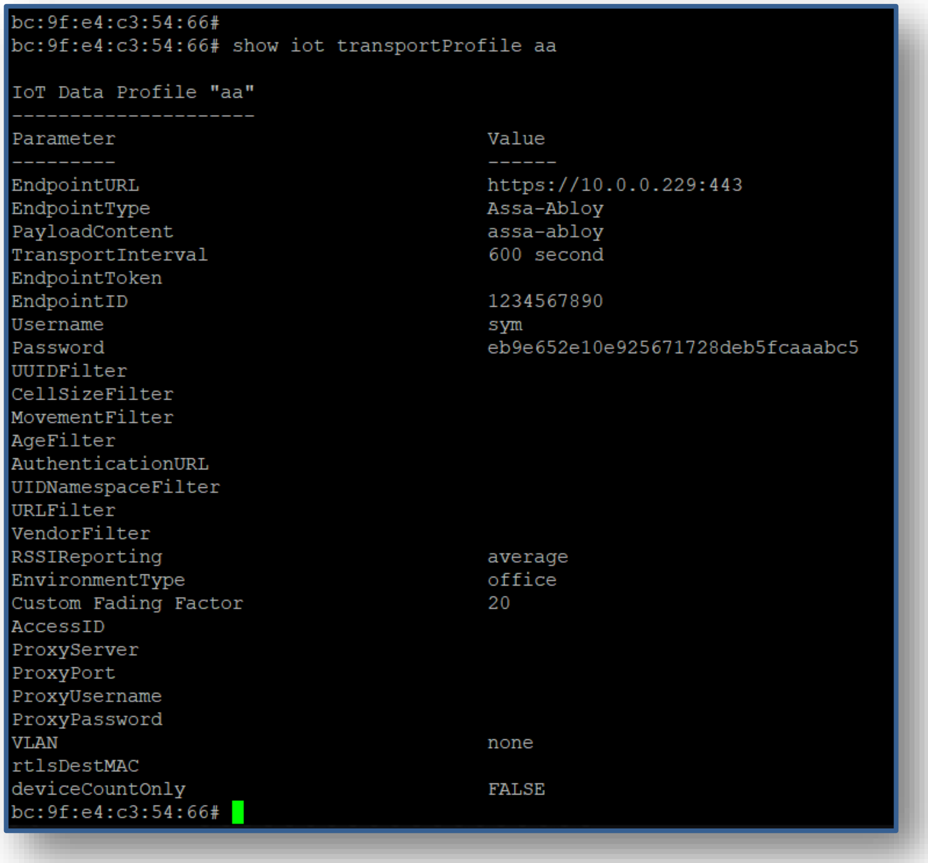
Output

#show iot transportProfile



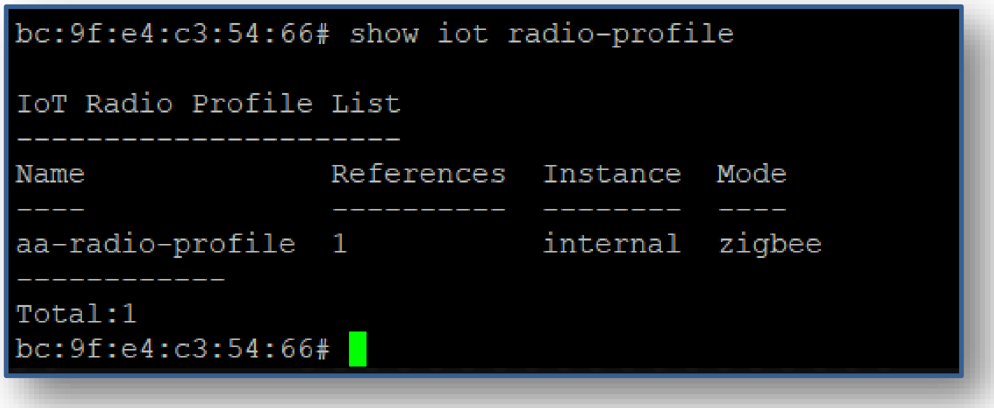
```
bc:9f:e4:c3:54:66# show iot radio-profile

IoT Radio Profile List
----------------------
Name                References  Instance  Mode
----                ----------  --------  ----
aa-radio-profile  1           internal  zigbee
------------
Total:1
bc:9f:e4:c3:54:66#
```

#show iot transportProfile aa



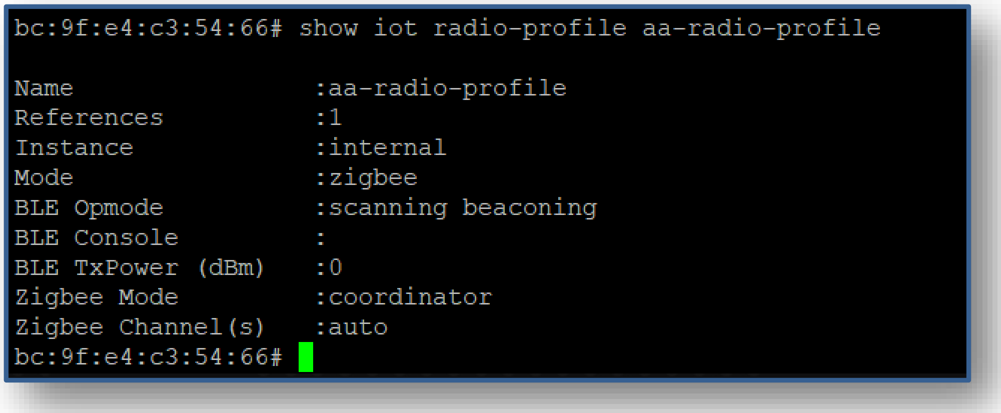
```
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show iot transportProfile aa

IoT Data Profile "aa"
---------------------
Parameter                                Value
---------                                ------
EndpointURL                              https://10.0.0.229:443
EndpointType                             Assa-Abloy
PayloadContent                           assa-abloy
TransportInterval                        600 second
EndpointToken
EndpointID                               1234567890
Username                                 sym
Password                                 eb9e652e10e925671728deb5fcaaabc5
UUIDFilter
CellSizeFilter
MovementFilter
AgeFilter
AuthenticationURL
UIDNamespaceFilter
URLFilter
VendorFilter
RSSIReporting                            average
EnvironmentType                          office
Custom Fading Factor                     20
AccessID
ProxyServer
ProxyPort
ProxyUsername
ProxyPassword
VLAN                                     none
rtlsDestMAC
deviceCountOnly                          FALSE
bc:9f:e4:c3:54:66#
```

#show iot radio-profile



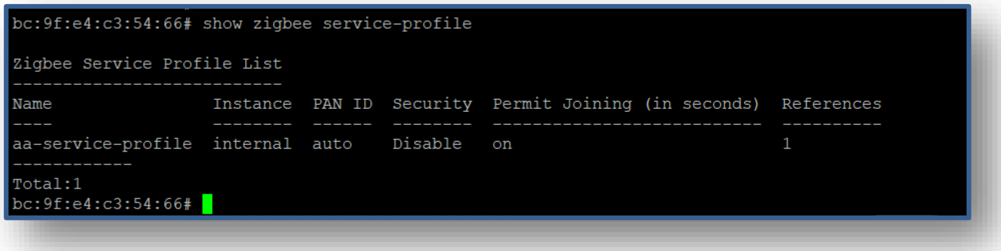
```
bc:9f:e4:c3:54:66# show iot radio-profile

IoT Radio Profile List
----------------------
Name              References  Instance  Mode
----              ----------  --------  ----
aa-radio-profile  1           internal  zigbee
------------
Total:1
bc:9f:e4:c3:54:66#
```

#show iot radio-profile aa-radio-profile



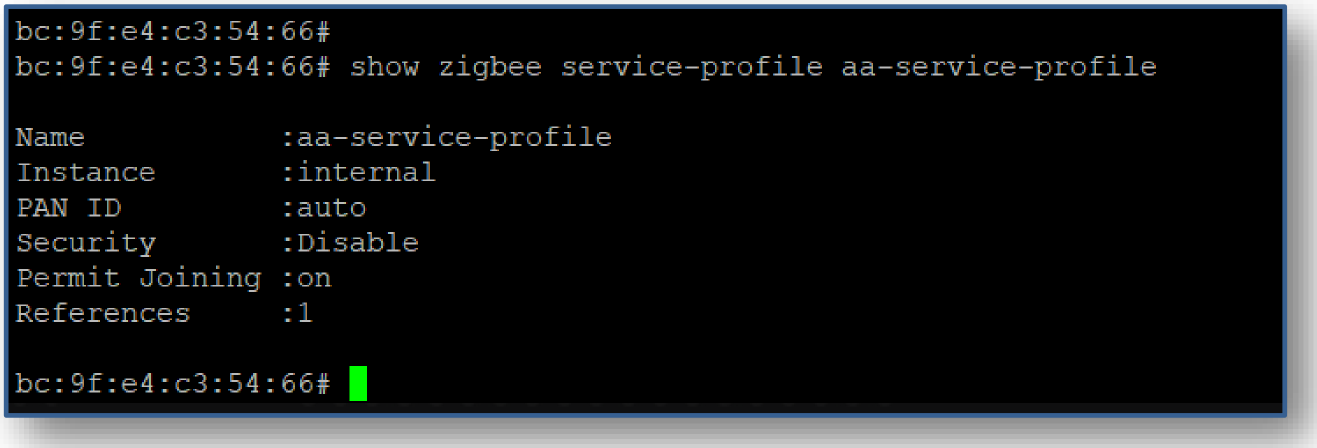
```
bc:9f:e4:c3:54:66# show iot radio-profile aa-radio-profile

Name                    :aa-radio-profile
References              :1
Instance                :internal
Mode                    :zigbee
BLE Opmode              :scanning beaconing
BLE Console             :
BLE TxPower (dBm)       :0
Zigbee Mode             :coordinator
Zigbee Channel(s)       :auto
bc:9f:e4:c3:54:66#
```

#show zigbee service-profile



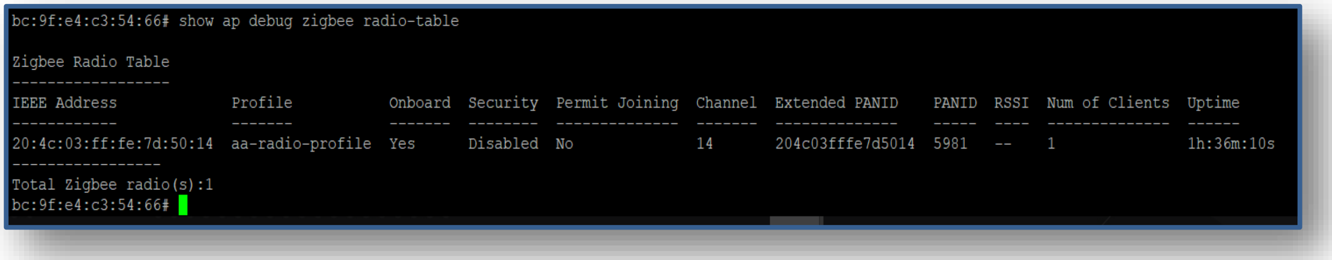
```
bc:9f:e4:c3:54:66# show zigbee service-profile

Zigbee Service Profile List
---------------------------
Name                 Instance  PAN ID  Security  Permit Joining (in seconds)  References
----                 --------  ------  --------  ---------------------------  ----------
aa-service-profile   internal  auto    Disable   on                           1
------------
Total:1
bc:9f:e4:c3:54:66#
```

#show zigbee service-profile aa-service-profile



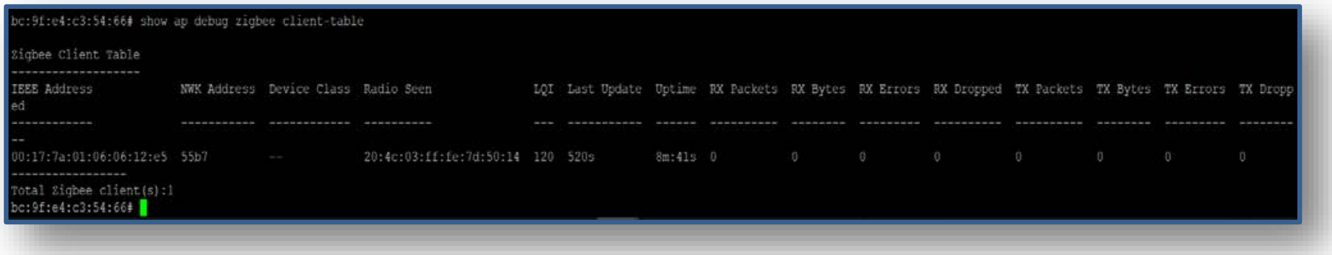
```
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show zigbee service-profile aa-service-profile

Name            :aa-service-profile
Instance        :internal
PAN ID          :auto
Security        :Disable
Permit Joining  :on
References      :1

bc:9f:e4:c3:54:66#
```

#show ap debug zigbee radio-table



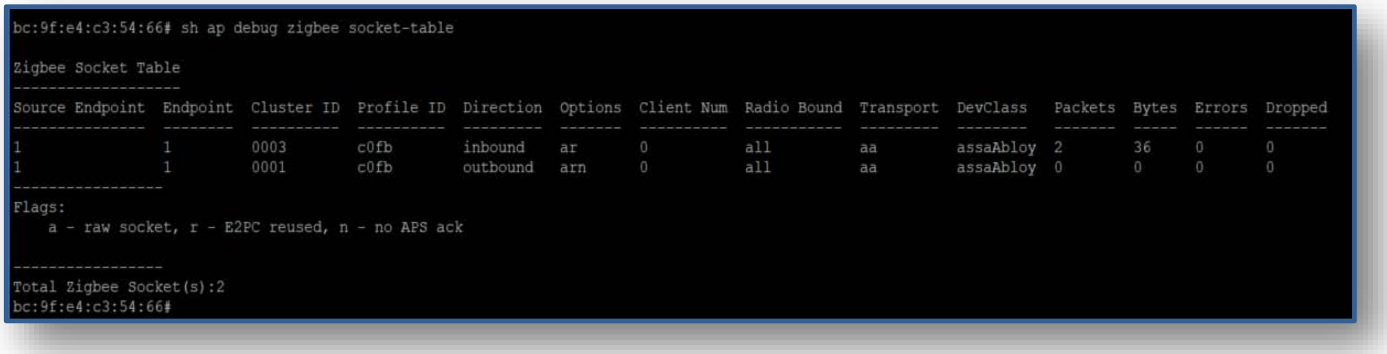
```
bc:9f:e4:c3:54:66# show ap debug zigbee radio-table

Zigbee Radio Table
------------------
IEEE Address             Profile            Onboard  Security  Permit Joining  Channel  Extended PANID    PANID  RSSI  Num of Clients  Uptime
------------             -------            -------  --------  --------------  -------  --------------    -----  ----  --------------  ------
20:4c:03:ff:fe:7d:50:14  aa-radio-profile   Yes      Disabled  No              14       204c03fffe7d5014  5981   --    1               1h:36m:10s
-----------------
Total Zigbee radio(s):1
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee client-table



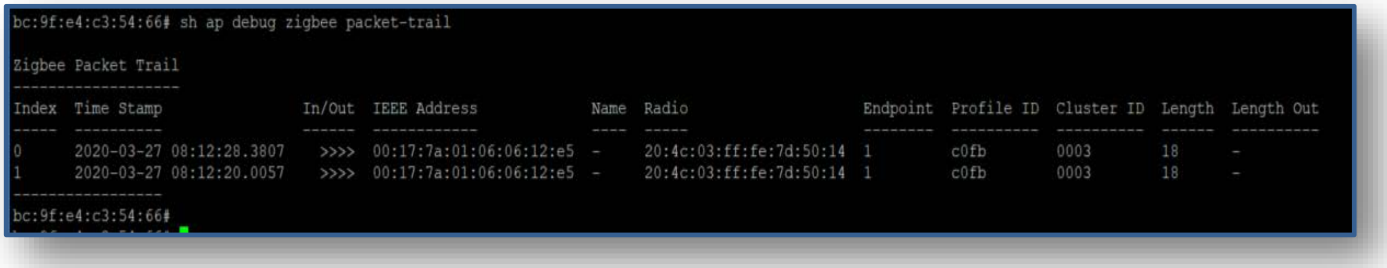
```
bc:9f:e4:c3:54:66# show ap debug zigbee client-table

Zigbee Client Table
-------------------
IEEE Address            NWK Address Device Class Radio Seen                LQI Last Update Uptime RX Packets RX Bytes RX Errors RX Dropped TX Packets TX Bytes TX Errors TX Dropp
ed
------------            ----------- ------------ ----------                --- ----------- ------ ---------- -------- --------- ---------- ---------- -------- --------- --------
--
00:17:7a:01:06:06:12:e5 55b7        --           20:4c:03:ff:fe:7d:50:14   120 520s        8m:41s 0          0        0         0          0          0        0         0
-----------------
Total Zigbee client(s):1
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee socket-table



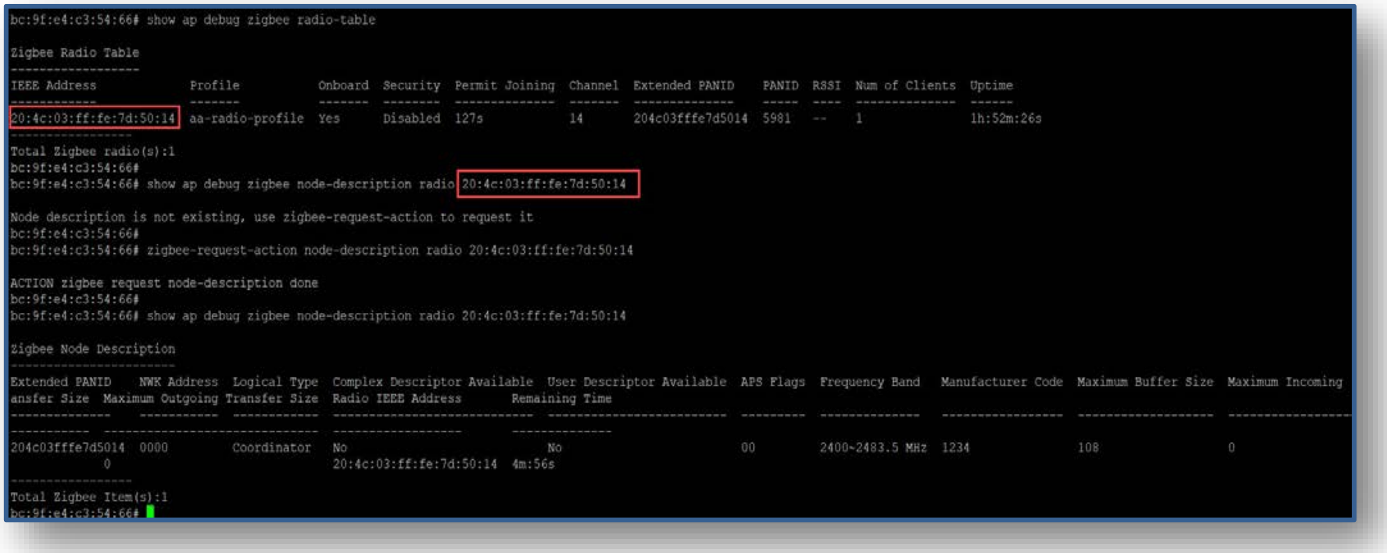
```
bc:9f:e4:c3:54:66# sh ap debug zigbee socket-table

Zigbee Socket Table
-------------------
Source Endpoint  Endpoint  Cluster ID  Profile ID  Direction  Options  Client Num  Radio Bound  Transport  DevClass   Packets  Bytes  Errors  Dropped
---------------  --------  ----------  ----------  ---------  -------  ----------  -----------  ---------  --------   -------  -----  ------  -------
1                1         0003        c0fb        inbound    ar       0           all          aa         assaAbloy  2        36     0       0
1                1         0001        c0fb        outbound   arn      0           all          aa         assaAbloy  0        0      0       0
-----------------
Flags:
    a - raw socket, r - E2PC reused, n - no APS ack

-----------------
Total Zigbee Socket(s):2
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee packet-trail



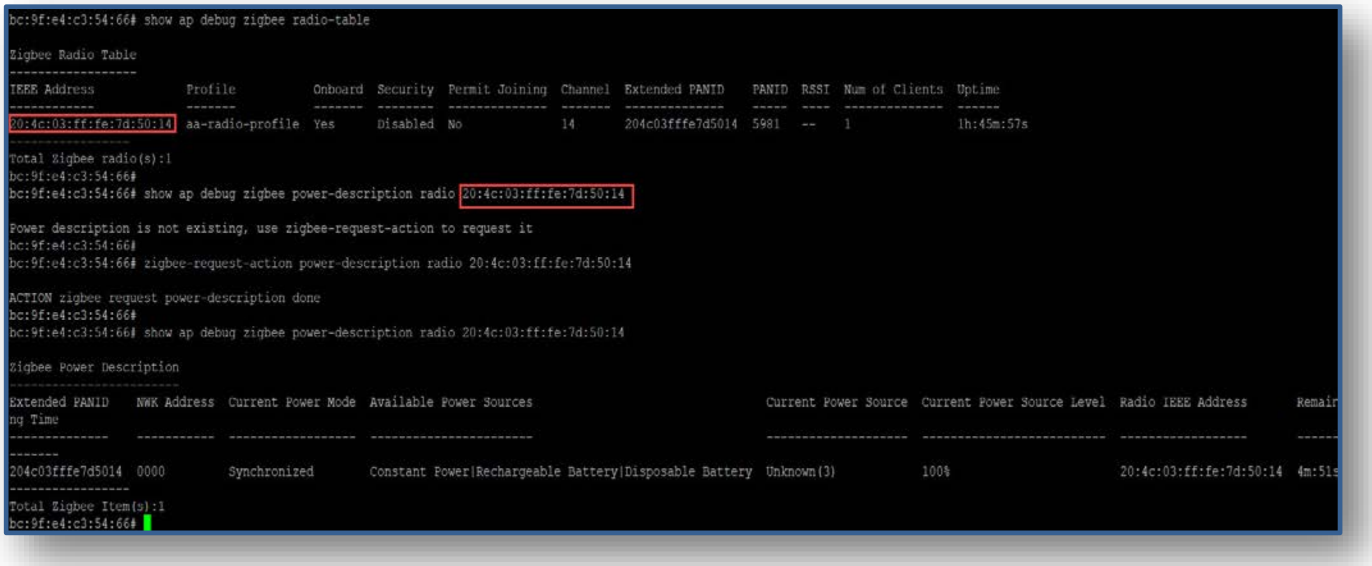
```
bc:9f:e4:c3:54:66# sh ap debug zigbee packet-trail

Zigbee Packet Trail
-------------------
Index  Time Stamp                   In/Out  IEEE Address             Name  Radio                    Endpoint  Profile ID  Cluster ID  Length  Length Out
-----  ----------                   ------  ------------             ----  -----                    --------  ----------  ----------  ------  ----------
0      2020-03-27 08:12:28.3807     >>>>    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:50:14  1         c0fb        0003        18      -
1      2020-03-27 08:12:20.0057     >>>>    00:17:7a:01:06:06:12:e5  -     20:4c:03:ff:fe:7d:50:14  1         c0fb        0003        18      -
-----------------
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee node-description radio <ap-zigbee-radio-mac>



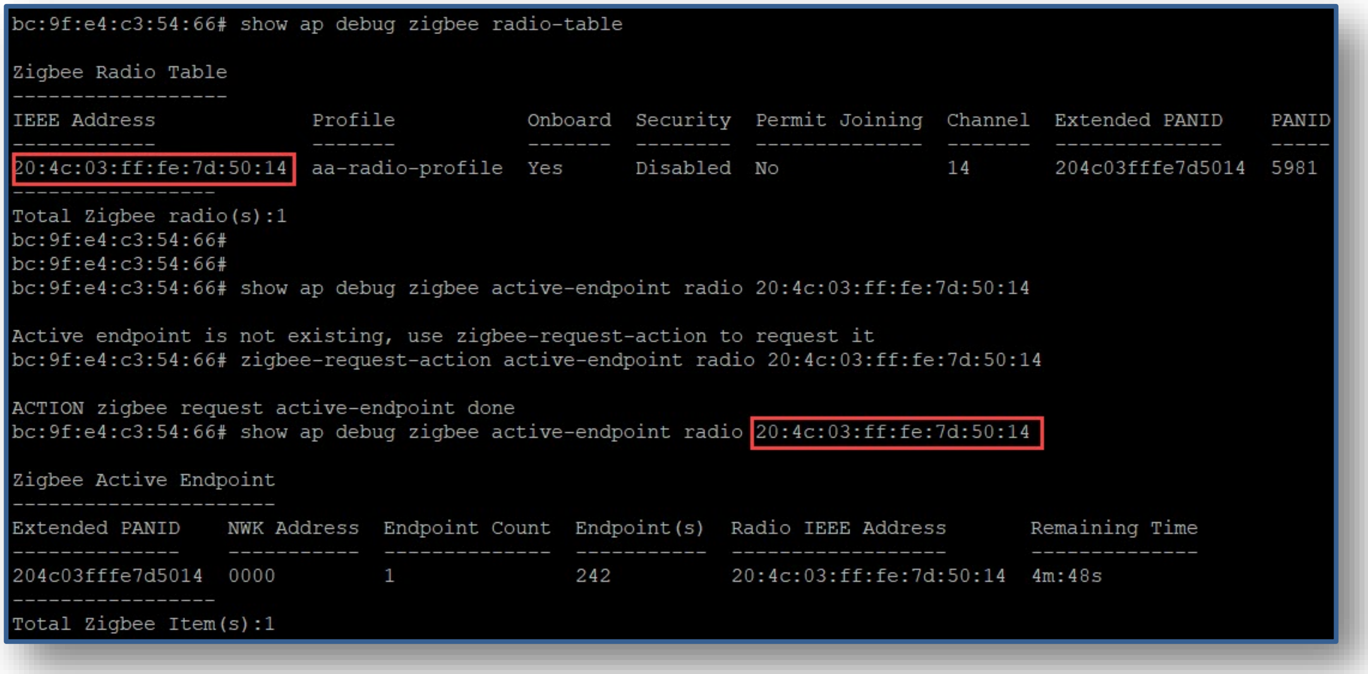
```
bc:9f:e4:c3:54:66# show ap debug zigbee radio-table

Zigbee Radio Table
------------------
IEEE Address             Profile           Onboard  Security  Permit Joining  Channel  Extended PANID    PANID  RSSI  Num of Clients  Uptime
------------             -------           -------  --------  --------------  -------  --------------    -----  ----  --------------  ------
20:4c:03:ff:fe:7d:50:14  aa-radio-profile  Yes      Disabled  127s            14       204c03fffe7d5014  5981   --    1               1h:52m:26s
-----------------
Total Zigbee radio(s):1
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show ap debug zigbee node-description radio 20:4c:03:ff:fe:7d:50:14

Node description is not existing, use zigbee-request-action to request it
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# zigbee-request-action node-description radio 20:4c:03:ff:fe:7d:50:14

ACTION zigbee request node-description done
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show ap debug zigbee node-description radio 20:4c:03:ff:fe:7d:50:14

Zigbee Node Description
-----------------------
Extended PANID    NWK Address  Logical Type  Complex Descriptor Available  User Descriptor Available  APS Flags  Frequency Band    Manufacturer Code  Maximum Buffer Size  Maximum Incoming
ansfer Size  Maximum Outgoing Transfer Size  Radio IEEE Address       Remaining Time
--------------    -----------  ------------  ----------------------------  -------------------------  ---------  --------------    -----------------  -------------------  -----------------
-----------  ------------------------------  ------------------       --------------
204c03fffe7d5014  0000         Coordinator   No                                                No     00         2400~2483.5 MHz   1234               108                  0
             0                               20:4c:03:ff:fe:7d:50:14  4m:56s
-----------------
Total Zigbee Item(s):1
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee power-description radio <ap-zigbee-radio-mac>



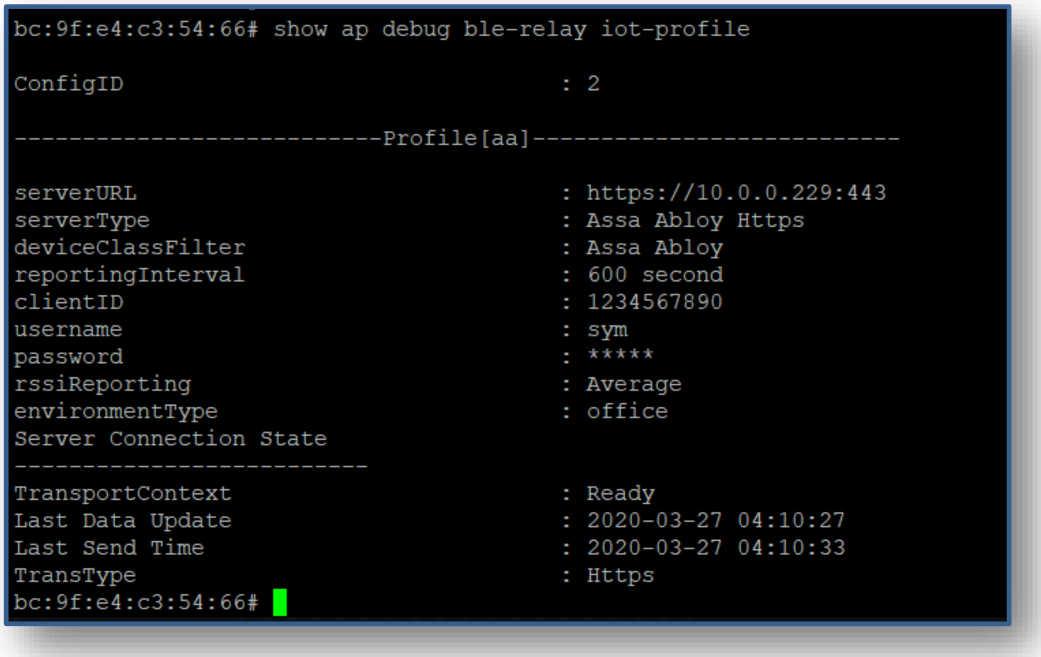
```
bc:9f:e4:c3:54:66# show ap debug zigbee radio-table

Zigbee Radio Table
------------------
IEEE Address             Profile           Onboard  Security  Permit Joining  Channel  Extended PANID    PANID  RSSI  Num of Clients  Uptime
------------             -------           -------  --------  --------------  -------  --------------    -----  ----  --------------  ------
20:4c:03:ff:fe:7d:50:14  aa-radio-profile  Yes      Disabled  No              14       204c03fffe7d5014  5981   --    1               1h:45m:57s
-----------------
Total Zigbee radio(s):1
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show ap debug zigbee power-description radio 20:4c:03:ff:fe:7d:50:14

Power description is not existing, use zigbee-request-action to request it
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# zigbee-request-action power-description radio 20:4c:03:ff:fe:7d:50:14

ACTION zigbee request power-description done
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show ap debug zigbee power-description radio 20:4c:03:ff:fe:7d:50:14

Zigbee Power Description
------------------------
Extended PANID    NWK Address  Current Power Mode  Available Power Sources                                Current Power Source  Current Power Source Level  Radio IEEE Address       Remair
ng Time
--------------    -----------  ------------------  -----------------------                                --------------------  --------------------------  ------------------       ------
-------
204c03fffe7d5014  0000         Synchronized        Constant Power|Rechargeable Battery|Disposable Battery  Unknown(3)            100%                        20:4c:03:ff:fe:7d:50:14  4m:51s
-----------------
Total Zigbee Item(s):1
bc:9f:e4:c3:54:66#
```

#show ap debug zigbee active-endpoint radio <ap-zigbee-radio-mac>



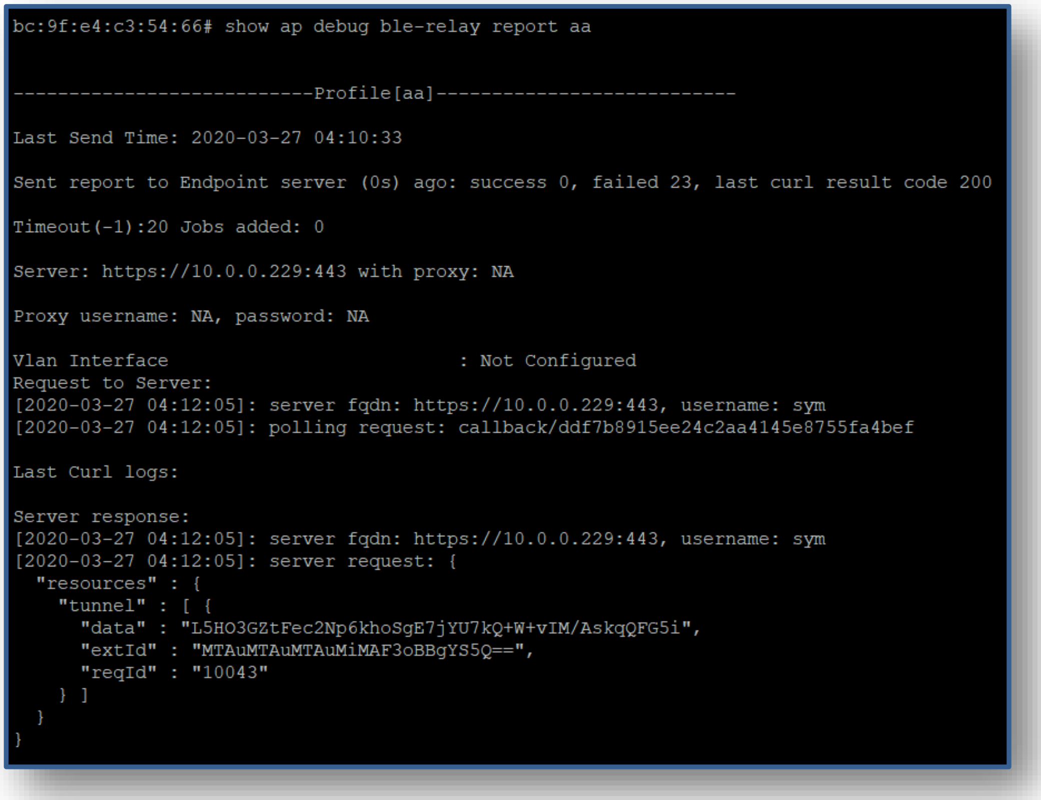
```
bc:9f:e4:c3:54:66# show ap debug zigbee radio-table

Zigbee Radio Table
------------------
IEEE Address             Profile           Onboard  Security  Permit Joining  Channel  Extended PANID    PANID
------------             -------           -------  --------  --------------  -------  --------------    -----
20:4c:03:ff:fe:7d:50:14  aa-radio-profile  Yes      Disabled  No              14       204c03fffe7d5014  5981
-----------------
Total Zigbee radio(s):1
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66#
bc:9f:e4:c3:54:66# show ap debug zigbee active-endpoint radio 20:4c:03:ff:fe:7d:50:14

Active endpoint is not existing, use zigbee-request-action to request it
bc:9f:e4:c3:54:66# zigbee-request-action active-endpoint radio 20:4c:03:ff:fe:7d:50:14

ACTION zigbee request active-endpoint done
bc:9f:e4:c3:54:66# show ap debug zigbee active-endpoint radio 20:4c:03:ff:fe:7d:50:14

Zigbee Active Endpoint
----------------------
Extended PANID    NWK Address  Endpoint Count  Endpoint(s)  Radio IEEE Address       Remaining Time
--------------    -----------  --------------  -----------  ------------------       --------------
204c03fffe7d5014  0000         1               242          20:4c:03:ff:fe:7d:50:14  4m:48s
-----------------
Total Zigbee Item(s):1
```

#show ap debug ble-relay iot-profile



```
bc:9f:e4:c3:54:66# show ap debug ble-relay iot-profile

ConfigID                                : 2

---------------------------Profile[aa]---------------------------

serverURL                               : https://10.0.0.229:443
serverType                              : Assa Abloy Https
deviceClassFilter                       : Assa Abloy
reportingInterval                       : 600 second
clientID                                : 1234567890
username                                : sym
password                                : *****
rssiReporting                           : Average
environmentType                         : office
Server Connection State
--------------------------
TransportContext                        : Ready
Last Data Update                        : 2020-03-27 04:10:27
Last Send Time                          : 2020-03-27 04:10:33
TransType                               : Https
bc:9f:e4:c3:54:66#
```

#show ap debug ble-relay report aa



```
bc:9f:e4:c3:54:66# show ap debug ble-relay report aa


---------------------------Profile[aa]---------------------------

Last Send Time: 2020-03-27 04:10:33

Sent report to Endpoint server (0s) ago: success 0, failed 23, last curl result code 200

Timeout(-1):20 Jobs added: 0

Server: https://10.0.0.229:443 with proxy: NA

Proxy username: NA, password: NA

Vlan Interface                          : Not Configured
Request to Server:
[2020-03-27 04:12:05]: server fqdn: https://10.0.0.229:443, username: sym
[2020-03-27 04:12:05]: polling request: callback/ddf7b8915ee24c2aa4145e8755fa4bef

Last Curl logs:

Server response:
[2020-03-27 04:12:05]: server fqdn: https://10.0.0.229:443, username: sym
[2020-03-27 04:12:05]: server request: {
  "resources" : {
    "tunnel" : [ {
      "data" : "L5HO3GZtFec2Np6khoSgE7jYU7kQ+W+vIM/AskqQFG5i",
      "extId" : "MTAuMTAuMTAuMiMAF3oBBgYS5Q==",
      "reqId" : "10043"
    } ]
  }
}
```

FAQs

What is the limit on the number of profiles user can configure?
Maximum of 10 transport profiles are supported per controller. One ap-group can be a part of 4 profiles maximum.

Any other specific dependencies?
User can only have 1 Meridian-Beacons-Management profile per ap-group.

What if I do not see my device class in the list?
We are constantly adding new device classes. Currently if you do not see your device class listed, it might not be supported.