Configuring Mobility

Configuring Mobility Controller

Configuring Converged Access Controllers

Creating Peer Groups, Peer Group Member, and Bridge Domain ID (CLI)

Before you begin

- On the mobility agent, you can only configure the IP address of the mobility controller.
- On the mobility controller, you can define the peer group and the IP address of each peer group member.

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility controller** **Example:** `Device(config)# wireless mobility controller`	Enables the mobility controller functionality on the device. This command is applicable only to the switch. The controller is by default a mobility controller.
Step 2	**wireless mobility controller peer-group** *SPG1* **Example:** `Device(config)# wireless mobility controller peer-group SPG1`	Creates a peer group named SPG1.

	Command or Action	Purpose
Step 3	**wireless mobility controller peer-group** *SPG1* **member ip** *member-ip-addr* **public-ip** *public-ip-addr* **Example:** `Device(config)# wireless mobility controller peer-group SPG1 member ip 10.10.20.2 public-ip 10.10.20.2`	Adds a mobility agent to the peer group. **Note** The `10.10.20.2` is the mobility agent's direct IP address. When NAT is used, use the optional public IP address to enter the mobility agent's NATed address. When NAT is not used, the public IP address is not used and the device displays the mobility agent's direct IP address.
Step 4	**wireless mobility controller peer-group** *SPG1* **member ip** *member-ip-addr* **public-ip** *public-ip-addr* **Example:** `Device(config)# wireless mobility controller peer-group SPG1 member ip 10.10.20.6 public-ip 10.10.20.6`	Adds another member to the peer group SPG1.
Step 5	**wireless mobility controller peer-group** *SPG2* **Example:** `Device(config)# wireless mobility controller peer-group SPG2`	Creates another peer group SPG2.
Step 6	**wireless mobility controller peer-group** *SPG2* **member ip** *member-ip-addr* **public-ip** *public-ip-addr* **Example:** `Device(config)# wireless mobility controller peer-group SPG2 member ip 10.10.10.20 public-ip 10.10.10.20`	Adds a member to peer group SPG2.
Step 7	**wireless mobility controller peer-group** *SPG1* **bridge-domain-id** *id* **Example:** `Device(config)# wireless mobility controller peer-group SPG1 bridge-domain-id 54`	(Optional) Adds a bridge domain to SPG1 used for defining the subnet-VLAN mapping with other SPGs.

Example

This example shows how to create peer group and add members to it:

```
Device(config)# wireless mobility controller
Device(config)# wireless mobility controller peer-group SPG1
Device(config)# wireless mobility controller peer-group SPG1
Device(config)# wireless mobility controller peer-group SPG1 member ip 10.10.20.2 public-ip
```

```
 10.10.20.2
Device(config)# wireless mobility controller peer-group SPG1 member ip 10.10.20.6 public-ip
 10.10.20.6
Device(config)# wireless mobility controller peer-group SPG2
Device(config)# wireless mobility controller peer-group SPG2 member ip 10.10.10.20 public-ip
 10.10.10.20
Device(config)# wireless mobility controller peer-group SPG1 bridge-domain-id 54
```

Configuring Local Mobility Group (CLI)

Configuration for wireless mobility groups and mobility group members where the mobility group is a group of MCs.

Before you begin

MCs can belong only to one mobility group, and can know MCs in several mobility groups.

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility group name** *group-name* **Example:** `Device(config)# wireless mobility group name Mygroup`	Creates a mobility group named **Mygroup**.
Step 2	**wireless mobility group member ip** *member-ip-addr* **public-ip** *public-ip-addr* **Example:** `Device(config)# wireless mobility group member ip 10.10.34.10 public-ip 10.10.34.28`	Adds a mobility controller to the **Mygroup** mobility group. **Note** When NAT is used, use the optional public IP address to enter the NATed IP address of the mobility controller.
Step 3	**wireless mobility group keepalive interval** *time-in-seconds* **Example:** `Device(config)# wireless mobility group keepalive interval 5`	Configures the interval between two keepalives sent to a mobility member.
Step 4	**wireless mobility group keepalive count** *count* **Example:** `Device(config)# wireless mobility group keepalive count 3`	Configures the keep alive retries before a member status is termed DOWN.

Example

```
Device(config)# wireless mobility group name Mygroup
Device(config)# wireless mobility group member ip 10.10.34.10 public-ip 10.10.34.28
Device(config)# wireless mobility group keepalive interval 5
Device(config)# wireless mobility group keepalive count 3
```

Adding a Peer Mobility Group (CLI)

Before you begin

MCs belong to only one group, and can know MCs in several groups.

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility group member ip** *member-ip-addr* **public-ip** *public-ip-addr* **group** *group-name* **Example:** `Device(config)# wireless mobility group member ip 10.10.10.24 public-ip 10.10.10.25 group Group2`	Adds the member as a peer MC in a different group than the **Mygroup**.

Configuring Optional Parameters for Roaming Behavior

Use this configuration to disable the sticky anchor. This command can also be used, if required, between all MA's and MC's where roaming is expected for the target SSID.

Procedure

	Command or Action	Purpose
Step 1	**wlan open21** **Example:** `Device(config)# wlan open20`	Configures a WLAN.
Step 2	**no mobility anchor sticky** **Example:** `Device(config-wlan)# no mobility anchor sticky`	Disables the default sticky mobility anchor.

Example

```
Device(config)# wlan open20
Device(config-wlan)# no mobility anchor sticky
```

Pointing the Mobility Controller to a Mobility Oracle (CLI)

Before you begin

You can configure a mobility oracle on a known mobility controller.

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility group member ip** *member-ip-addr* **group** *group-name* **Example:** `Device(config)# wireless mobility group member ip 10.10.10.10 group Group3`	Creates and adds a MC to a mobility group.
Step 2	**wireless mobility oracle ip** *oracle-ip-addr* **Example:** `Device(config)# wireless mobility oracle ip 10.10.10.10`	Configures the mobility controller as mobility oracle.

Example

```
Device(config)# wireless mobility group member ip 10.10.10.10 group Group3
Device(config)# wireless mobility oracle ip 10.10.10.10
```

Configuring Guest Controller

A guest controller is used when the client traffic is tunneled to a guest anchor controller in the demilitarized zone (DMZ). The guest client goes through a web authentication process. The web authentication process is optional, and the guest is allowed to pass traffic without authentication too.

Enable the WLAN on the mobility agent on which the guest client connects with the mobility anchor address of the guest controller.

On the guest controller WLAN, which can be Cisco 5500 Series WLC, Cisco WiSM2, or Cisco 5700 Series WLC, configure the IP address of the mobility anchor as its own IP address. This allows the traffic to be tunneled to the guest controller from the mobility agent.

Note With Cisco 5700 Series WLC as the guest anchor controller and Cisco 5500 Series WLC or Cisco WiSM2 as export foreign controller, the guest user role per user is not supported on the Cisco 5700 Series WLC.

Procedure

	Command or Action	Purpose
Step 1	**wlan** *wlan-id* **Example:** `Device(config)# wlan Mywlan1`	Creates a WLAN for the client.

	Command or Action	Purpose
Step 2	**mobility anchor** *guest-anchor-ip-addr* **Example:** `Device(config-wlan)# mobility anchor 10.10.10.2`	Enables the guest anchors (GA) IP address on the MA. **Note** To enable guest anchor on the mobility controller, you need not enter the IP address. Enter the **mobility anchor** command in the WLAN configuration mode to enable GA on the mobility controller.
Step 3	**client vlan** *vlan-name* **Example:** `Device(config-wlan)# client vlan gc_ga_vlan1`	Assigns a VLAN to the client's WLAN.
Step 4	**security open** **Example:** `Device(config-wlan)# security open`	Assigns a security type to the WLAN.

Example

```
Device(config)# wlan Mywlan1
Device(config-wlan)# mobility anchor 10.10.10.2
Device(config-wlan)# client vlan gc_ga_vlan1
Device(config-wlan)# security open
```

Configuring Guest Anchor

Procedure

	Command or Action	Purpose
Step 1	**wlan Mywlan1** **Example:** `Device(config)# wlan Mywlan1`	Creates a wlan for the client.
Step 2	**mobility anchor <guest-anchors-own-ip-address>** **Example:** `Device(config-wlan)# mobility anchor 10.10.10.2`	Enables the guest anchors IP address on the guest anchor (GA). The GA assigns its own address on itself.
Step 3	**client vlan<vlan-name>** **Example:** `Device(config-wlan)# client vlan gc_ga_vlan1`	Assigns a vlan to the clients wlan.

	Command or Action	Purpose
Step 4	**security open** **Example:** `Device(config-wlan)# security open`	Assigns a security type to the wlan.

Example

```
Device(config)# wlan Mywlan1
Device(config-wlan)# mobility anchor 10.10.10.2
Device(config-wlan)# client vlan gc_ga_vlan1
Device(config-wlan)# security open
```

Configuring Mobility Agent

Configuring Mobility Agent by Pointing to Mobility Controller (CLI)

Before you begin

- By default, the switches are configured as mobility agents.
- Your network must have at least one mobility controller and the network connectivity with the mobility controller must be operational.
- You cannot configure mobility from the mobility agent. On the mobility agent, you can configure only the IP address of the mobility controller to download the SPG configuration.
- On the mobility agent, you can either configure the mobility controller address to point to an external mobility agent, or enable the mobility controller function.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless management interface vlan 21** **Example:** `Device (config)# wireless management interface vlan 21`	Enables the wireless functionality on the device and activates the mobility agent function. This ensures the APs have a place to terminate the CAPWAP tunnel.

Example

This example shows how to add a mobility agent into the mobility group by pointing it to a mobility controller:

```
Device(config)# wireless management interface vlan 21
```

Configuring the Mobility Controller for the Mobility Agent (CLI)

Procedure

	Command or Action	Purpose

Step 1	**wireless mobility controller**
	Example:
	`Device (config)# wireless mobility controller`
	`Mobility role changed to Mobility Controller.`
	`Please save config and reboot the whole stack.`
		Enables the mobility function on the device.
		Note After you enter this command, save the configuration and reboot the device for the mobility controller function to take effect.

Step 2	**wireless mobility controller ip** *ip-addr* **Example:** `Device (config)# wireless mobility controller ip 10.10.21.3`	Specifies the mobility controller to which the mobility agent relates. **Note** If a mobility agent is configured and the mobility controller exists on a different device, configure the SPG on the mobility controller to ensure the mobility agent functions properly.

What to do next

After you add a mobility controller role to the mobility agent, you can configure optional parameters on the mobility agent.

Adding a Mobility Controller Role to the Mobility Agent

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility controller ip** `10.10.21.3` **Example:**	Converts the mobility agent to a mobility controller.

	Command or Action	Purpose
	`Device(config)# wireless mobility controller ip 10.10.21.3`	

Example

This example shows how to add the mobility controller role to a mobility agent:

```
Device(config)# wireless mobility controller ip 10.10.21.3
Mobility role changed to Mobility Controller.
Please save config and reboot the whole stack.
```

Configuring Optional Parameters on a Mobility Agent (CLI)

This section shows how to configure load-balancing on a device.

- By default, the load-balancing is enabled and it cannot be disabled.
- The device supports a maximum of 2000 clients and the default threshold value is fifty percent of client max load.
- When the device reaches its threshold, it redistributes the new clients load to other mobility agents in the same SPG, if their client load is lower.

Procedure

	Command or Action	Purpose
Step 1	**wireless mobility load-balance threshold** *threshold-value* **Example:** `Device(config)# wireless mobility load-balance threshold 150`	Configures the threshold that triggers load-balancing.

Managing Mobility Agents with Mobility Controller

Managing Mobility Agents with Mobility Controller

The Mobility Controller managing Mobility Agent feature pushes the wireless and common configurations from the Mobility Controller (MC) to the Mobility Agents (MAs). This helps you to easily configure, monitor, and troubleshoot all the MAs from the MC. An MC can support up to 16 MAs. Most of the wireless and common configurations such as AAA, ACL, and so on are generally the same across all the switches.

Restrictions for Managing Mobility Agents with Mobility Controller

- The MC and MAs can become unsynchronized when a new MA joins the MC, and this MA is centrally managed, and when an MA is moved from one MC to another MC.
- When an MA is in centralized mode, the globally managed configuration is disabled and the rest of the configurations and monitoring are available on the Web GUI.
- This feature is not supported on Cisco Prime Infrastructure.
- When the MC detects an MA to be out of sync, the MA is forced to reload and then resync the entire configuration from the MC after coming up again.
- QoS configuration is not pushed from the MC to an MA.
- The MC pushes all the configurations to all the centrally managed MAs. It is not possible to select a subset of the configurations and then push to a particular group of MAs instead of all the MAs.
- L3 roaming is not supported because WLAN configuration is pushed from the MC.
- MAs using different software versions in the subdomain either in the same Switch Peer Group (SPG) or in a different SPG are not supported.
- If Catalyst 3850 and 3650 Switch act as an MC in a mobility subdomain, the MC and MAs only run Polaris release 16.2 and Polaris release16.1.1.
- Since only Catalyst 3850 Switches are allowed in the mobility subdomain there cannot be a 5760 added to a MC when Polaris version acts as MA.
- In 16.2 Denali release, the Cisco 3850 controller supports a maxiumum of 50 APs in MA or MC. In MC-MA scenario, a maximum of 100 APs are supported, which includes APs of both MA and MC.

Feature History

Release	Remarks
Cisco IOS XE Denali 16.2.1	The feature was introduced on the Cisco Catalyst 3850 and Cisco Catalyst 3650 Series Switches.

For a complete list of commands that are synchronized between the MC and the MA, see *MC Managing MA - List of Commands Synchronized Between MC and MA* at:

http://www.cisco.com/c/en/us/td/docs/wireless/controller/mc-ma/mc-ma-sync.html.

Information About Managing Mobility Agents with Mobility Controller

A Mobility Controller (MC) can have both centrally managed and noncentrally managed Mobility Agents (MAs) at the same time. A centrally managed MA receives a set of configurations that are configured on the MC. A noncentrally managed MA does not receive any configuration from the MC. While an MA is being centrally managed, you can not modify any of the configurations that are pushed from the MC to an MA.

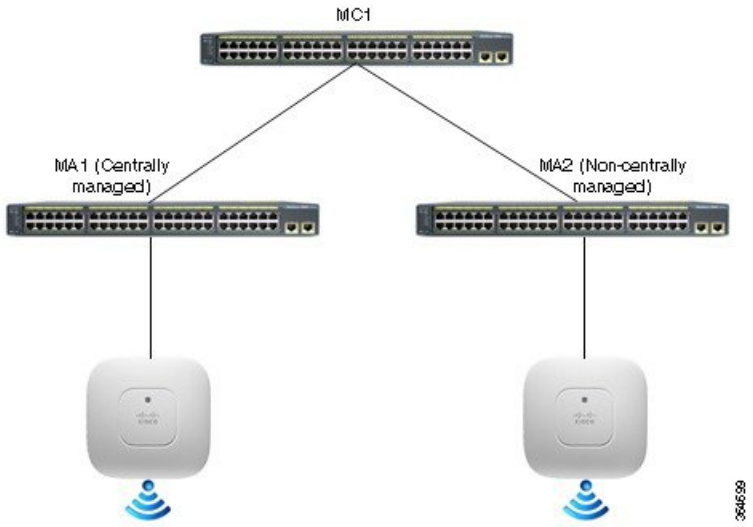
The MC pushes all the relevant configurations over the existing Control and Provisioning of Wireless Access Point (CAPWAP) tunnel to all the centrally managed MAs. The MC also pushes incremental configurations, if any, to the MAs.

Note Before using this feature, you must have the Day 0 configuration that is required to bring up the CAPWAP tunnel between the MC and an MA.

The following configurations are sent to the MAs:

- Common configuration—This is the configuration, that is shared between wired and wireless configurations such as the security configuration, for example, authentication, authorization, and accounting (AAA).
- Wireless configurations—All wireless configurations.

Figure 1: MC Managing MAs



Distributed Mode and Centralized Mode

Distributed mode refers to configurations that are required to be performed explicitly on all the MAs.

Centralized mode refers to configurations where wireless and common configurations are pushed from the MC to the MAs. The following table lists the differences between the Distributed mode and the Centralized mode.

Table: Differences Between Distributed Mode and Centralized Mode

Distributed Mode	Centralized Mode
Configurations on the MC	Configurations on the MC

Distributed Mode	Centralized Mode
• MA to MC Mobility Peering Configuration • Wireless LAN • Wireless QoS Policies • Wireless Security ACL • AAA Global Configurations • Location • Cisco CleanAir, Radio Resource Management (RRM), Client Link • Global and Per-AP Configuration	• MA to MC Mobility Peering Configuration • Wireless LAN • Wireless Security ACL • AAA Global Configurations • Location • Cisco CleanAir, RRM, Client Link • Global and Per-AP Configuration
Configurations on an MA	Configurations on an MA
• MA to MC Mobility Peering Configuration • Wireless LAN • Wireless QoS Policies • Wireless Security ACL • AAA Global Configurations • Location • Cisco CleanAir, RRM, Client Link • Global and Per-AP Configuration	• MA to MC Mobility Peering Configuration • Wireless QoS Policies

Restrictions for Managing Mobility Agents with Mobility Controller

- The MC and MAs can become unsynchronized when a new MA joins the MC, and this MA is centrally managed, and when an MA is moved from one MC to another MC.
- When an MA is in centralized mode, the globally managed configuration is disabled and the rest of the configurations and monitoring are available on the Web GUI.
- This feature is not supported on Cisco Prime Infrastructure.
- When the MC detects an MA to be out of sync, the MA is forced to reload and then resync the entire configuration from the MC after coming up again.
- QoS configuration is not pushed from the MC to an MA.
- The MC pushes all the configurations to all the centrally managed MAs. It is not possible to select a subset of the configurations and then push to a particular group of MAs instead of all the MAs.
- L3 roaming is not supported because WLAN configuration is pushed from the MC.

- MAs using different software versions in the subdomain either in the same Switch Peer Group (SPG) or in a different SPG are not supported.
- If Catalyst 3850 and 3650 Switch act as an MC in a mobility subdomain, the MC and MAs only run Polaris release 16.2 and Polaris release16.1.1.
- Since only Catalyst 3850 Switches are allowed in the mobility subdomain there cannot be a 5760 added to a MC when Polaris version acts as MA.
- In 16.2 Denali release, the Cisco 3850 controller supports a maxiumum of 50 APs in MA or MC. In MC-MA scenario, a maximum of 100 APs are supported, which includes APs of both MA and MC.

Configuring MC Managing MA (CLI)

The following procedure shows how to configure Managing Mobility Agents with Mobility Controller:

Procedure

Step 1 Configuring MC:

a) Configure a wireless management interface by entering this command:

```
Device(config)# wireless management interface vlan vlan-id
```

b) Configure a switch peer group by entering this command:

```
Device(config)# wireless mobility controller peer-group spg-name
```

c) Add an MA to the SPG and configure it to be centrally managed by entering this command: (Use only centralized option)

```
Device(config)# wireless mobility controller peer-group spg-name member ip ip-addr mode
 centralized
```

Step 2 Configuring MA:

a) Specify the IP address of the MC by entering this command:

```
Device(config)# wireless mobility controller ip mc-ip-addr
```

b) Configure the wireless management interface by entering this command:

```
Device(config)# wireless management interface vlan vlan-id
```

Step 3 Configuring Centralized Mode:

a) From the MC, you can see the status of the MA by entering this command:

```
Device(config)# show wireless mobility summary

Mobility Controller Summary:

Mobility Role                                 : Mobility Controller
Mobility Protocol Port                        : 16666
Mobility Group Name                           : default
Mobility Oracle IP Address                    : 0.0.0.0
DTLS Mode                                     : Enabled
Mobility Domain ID for 802.11r                : 0xac34
Mobility Keepalive Interval                   : 10
Mobility Keepalive Count                      : 3
```

```
Mobility Control Message DSCP Value        : 48
Mobility Domain Member Count               : 1

IP         Public IP        Link   Status     Centralized(Cfgd : Running)
-------------------------------------------------------------------------------
1.1.1.1     1.1.1.1          UP   : UP         Enabled        Enabled
3.3.3.1     3.3.3.1          DOWN : DOWN       Enabled        Enabled
```

The following table shows the details of the Centralized Mode (both Configured and Running) in relation to the example above.

Table 1: Centralized Mode (Configured and Running)

Sl. No.	Centralized Mode Configuration	Centralized Mode Running	What it Means
1.	Disabled	Disabled	The MA is not configured as centrally managed on the MC.
2.	Enabled	Disabled	The MA is configured as centrally managed on the MC, but the tunnel to the MA is still down, or the MA is yet to acknowledge the message from the MC (the message in which the MC informs the MA that it is centrally managed).
3.	Enabled	Enabled	The MA is configured as centrally managed on the MC and the MA is running in Centrally Managed mode.
4.	Disabled	Enabled	Not applicable.

b) You can see all the MAs that have been configured on the MC irrespective of the SPG and irrespective of whether they are centrally managed or not, by entering this command:

```
Device(config)# show cmm member-table

CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 16

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.155
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 16
Last rcvd cfg rev on MA = 16
Tunnel State            = Up
```

```
Status                  = CMM_MEMBER_STATUS_IN_SYNC
Last sent cfg rev to MA = 16
Last sent cfg timestamp = 1427826323 sec 936009397 nsec
----------------


Members: No. of MAs configured on the MC
System Rev No on MC: What version number the MC is at

Entry
```

c) To see the configurations that were executed on an MC and buffered in the CMM agent, enter this command:

```
Device(config)# show cmm config

Current version number: 17
To sync and save configuration to Mobility Agents execute: "write memory"

Config commands present in the buffer:
access-list 1 permit any
wlan MCMA_Demo 4 MCMA_Demo
client vlan 22
no security wpa
no security wpa akm dot1x
no security wpa wpa2
no security wpa wpa2 ciphers aes
no shutdown
```

Note The configuration from the MC is synchronized with MAs only after the **write memory** command is run on the MC.

Step 4 To execute the commands on an MA remotely from the MC, use this command. For example, you can enter this command on the MC to see if the client has reached the uptime.

```
Device(config)# remote command 1.1.1.1 show wcdb da all
Total Number of Wireless Clients = 1
Clients Waiting to Join = 0
Local Clients = 0
Anchor Clients = 1
Foreign Clients = 0
MTE Clients = 0
Mac Address VlanId IP Address Src If Auth Mob
---------- --------- ------------- -------- ----- -------
ec55.f9c6.35c3 22 53.1.1.2 0x00D19B00000001C5 RUN ANCHOR
```

Step 5 To log in remotely to an MA from the MC, use this command:

```
DeviceControllerDevice(config)# remote login 1.1.1.1

Trying Switch ...
Entering CONSOLE for Switch
Type "^C^C^C" to end this session


User Access Verification

Password:
MA1>en
Password:
```

```
MA1#
```

Configuration of WLAN on the MC

This following procedure shows how to create a WLAN on the MC and synchronize the WLAN configuration with centrally managed MCs.

Procedure

Step 1 On the MC, create a WLAN named MCMA_Demo by entering this command:

```
Device(config)# wlan MCMA_Demo 1 MCMA_Demo
Device(config-wlan)# exit
Device(config)# exit
```

Step 2 Enter this command to check the configuration:

```
Device(config)# show cmm config

Current version number: 3
To sync and save configuration to Mobility Agents execute: "write memory"

Config commands present in the buffer:
wlan MCMA_Demo 1 MCMA_Demo
exit
```

Step 3 Enter this command to check the number of MAs that are configured to be centrally managed:

```
Device(config)# show cmm member-table

CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 2

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 2
Last rcvd cfg rev on MA = 2
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_IN_SYNC
Last sent cfg rev to MA = 2
Last sent cfg timestamp = 1432843797 sec 57656031 nsec
----------------
```

Step 4 See the WLAN details by entering this command:

```
Device(config)# show wlan summary

Number of WLANs: 1
```

```
WLAN  Profile Name    SSID          VLAN Status
-----------------------------------------------------
1        MCMA_Demo    MCMA_Demo       1    DOWN
```

Step 5 Save the configuration by entering this command:

```
Device(config)# write memory

Building configuration...
Compressed configuration from 7612 bytes to 3409 bytes[OK]
```

Step 6 Check the synchronization status on an MA by entering this command:

```
Device(config)# show cmm member-table

CMM Member Table
---------------
Total No Of Members = 1
System Rev No on MC = 3

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 2
Last rcvd cfg rev on MA = 2
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_STALE
Last sent cfg rev to MA = 3
Last sent cfg timestamp = 1432847325 sec 107200589 nsec
----------------
```

Step 7 On the MA, enter the following command to verify if the WLAN that was created in the MC is now synchronized with the MA:

```
Device(config)# show wlan summary

Number of WLANs: 1

WLAN  Profile Name    SSID           VLAN Status
-----------------------------------------------------
1    MCMA_Demo     MCMA_Demo          1    DOWN
```

Example:

Example logs showing how multiple configurations are synchronized

The following example shows output of cmm configuration:

The following example shows output of to view the cmm member-table:

The following example shows output of to view WLAN Summary:

The following example shows output of to write memory:

The following example shows output of WLAN summary:

The following example shows output of cmm member-table:

The following example shows output of cmm member-table:

The following example shows output of cmm member-table:

The following example shows output of configuration MA:

The following example shows output of WLAN summary:

The following example shows output of configuring cmm:

The following example shows output of view and run the WLAN:

The following example shows output of view and run the WLAN open:

```
MC#show cmm config
Current version number: 4
To sync and save configuration to Mobility Agents execute: "write memory"

Config commands present in the buffer:
wlan open 2 open
assisted-roaming dual-list
assisted-roaming neighbor-list
broadcast-ssid
ccx aironet-iesupport
channel-scan defer-priority 4
client association limit ap 0
client association limit radio 0
client vlan default
exclusionlist
exclusionlist timeout 60
ip access-group web none
mac-filtering test
mobility anchor sticky
radio all
security wpa
security wpa akm dot1x
security wpa wpa2
security wpa wpa2 ciphers aes
security dot1x authentication-list test
```

```
security dot1x encryption 104
security ft over-the-ds
security ft reassociation-timeout 20
security static-wep-key authentication open
security tkip hold-down 60
security web-auth authentication-list test2
security web-auth parameter-map test3
service-policy client input un
service-policy client output un
service-policy input unk
service-policy output unk
session-timeout 1800
no shutdown
exit
```

```
To view cmm member-table:
MC#show cmm member-table
CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 3

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 3
Last rcvd cfg rev on MA = 3
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_IN_SYNC
Last sent cfg rev to MA = 3
Last sent cfg timestamp = 1433441315 sec 669464681 nsec
----------------
```

```
To view WLAN summary:
MC#show wlan summary

Number of WLANs: 2

WLAN Profile Name                    SSID                           VLAN Status
------------------------------------------------------------------------------------
1    test                            test                           1    DOWN
2    open                            open                           1    UP
```

```
To write memory:
MC#write mem
Building configuration...
Compressed configuration from 7972 bytes to 3619 bytes[OK]


MC#show wlan summary

Number of WLANs: 2

WLAN Profile Name                    SSID                           VLAN Status
```

```
--------------------------------------------------------------------------------
1    test                              test                              1    DOWN
2    open                              open                              1    UP

To view cmmm config:
MC#show cmm config
Current version number: 4
To sync and save configuration to Mobility Agents execute: "write memory"

Config commands present in the buffer:


MC#show cmm member-table
CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 4

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 3
Last rcvd cfg rev on MA = 3
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_STALE
Last sent cfg rev to MA = 4
Last sent cfg timestamp = 1433488804 sec 349065646 nsec
----------------


MC#show cmm member-table
CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 4

entry 0
--------
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 3
Last rcvd cfg rev on MA = 3
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_STALE
Last sent cfg rev to MA = 4
Last sent cfg timestamp = 1433488812 sec 349323943 nsec
----------------


MC#show cmm member-table
CMM Member Table
----------------
Total No Of Members = 1
System Rev No on MC = 4

entry 0
--------
```

```
entry_status            = In use
ip_addr                 = 10.5.84.12
SPG Name                = SPG1
Centrally Managed       = True
Applied Cfg rev on MA   = 4
Last rcvd cfg rev on MA = 4
Tunnel State            = Up
Status                  = CMM_MEMBER_STATUS_IN_SYNC
Last sent cfg rev to MA = 4
Last sent cfg timestamp = 1433488820 sec 349544632 nsec
----------------
MC#
```

```
To view the cmm configuration
MA21#show cmm config
Current version number: 3
Centrally Managed: True
```

```
MA21#show wlan summary

Number of WLANs: 1

WLAN Profile Name                    SSID                           VLAN Status
-------------------------------------------------------------------------------
1    test                            test                           1    DOWN

MA21#
Building configuration...

*Jun  5 07:21:18.295: %SYS-5-CONFIG_I: Configured from console by vty1
*Jun  5 07:21:18.314: %CMM-6-CONFIG_SYNC_SAVE_MSG: Saving config rev#4 received
from Mobility Controller.Compressed configuration from 13033 bytes to 4340 bytes[OK]
```

```
MA21#show cmm config
Current version number: 4
Centrally Managed: True
MA21#show wlan summary

Number of WLANs: 2

WLAN Profile Name                    SSID                           VLAN Status
-------------------------------------------------------------------------------
1    test                            test                           1    DOWN
2    open                            open                           1    UP
```

```
MA21#show run wlan
wlan test 1 test
 shutdown
wlan open 2 open
 assisted-roaming dual-list
 assisted-roaming neighbor-list
 ip access-group web none
 mac-filtering test
 security dot1x authentication-list test
 security web-auth authentication-list test2
 security web-auth parameter-map test3
 service-policy client input un
```

```
 service-policy client output un
 service-policy input unk
 service-policy output unk
 no shutdown
MA21#



To view and run the WLAN open
MA21#show run wlan open
wlan open 2 open
 assisted-roaming dual-list
 assisted-roaming neighbor-list
 ip access-group web none
 mac-filtering test
 security dot1x authentication-list test
 security web-auth authentication-list test2
 security web-auth parameter-map test3
 service-policy client input un
 service-policy client output un
 service-policy input unk
 service-policy output unk
 no shutdown
MA21#
MA21#
```