



Multicast Domain Name System

- [mDNS Gateway, on page 1](#)
- [Location based service filtering, on page 27](#)

mDNS Gateway

A mDNS gateway is an Apple service discovery protocol that

- listens for multicast DNS (mDNS) service announcements and queries from local domains with the use of mDNS service records
- caches and forwards mDNS advertisements between segmented Layer 2 networks or subnets, and
- enables devices and services to be discovered even when clients and sources reside in different network segments.

The Bonjour protocol operates using service announcements and queries. Each query or advertisement is sent to the Bonjour multicast address IPv4 224.0.0.251 or IPv6 FF02::FB. The protocol uses mDNS on UDP port 5353.

The address used by the Bonjour protocol is a link-local multicast address and is therefore only forwarded to the local Layer 2 network. Because multicast DNS is limited to a Layer 2 domain, a client must be part of the same Layer 2 domain to discover a service. However, this requirement is not always practical in large-scale deployments or enterprises.

To address this issue, the Cisco Catalyst 9800 Series Wireless Controller acts as a Bonjour Gateway. The controller then listens for Bonjour services and caches these Bonjour advertisements (AirPlay, AirPrint, and so on) from the source or host. For example, Apple TV responds to Bonjour clients when requested for a service. This allows sources and clients to exist in different subnets.

By default, the mDNS gateway is disabled on the controller. To enable mDNS gateway functionality, you must explicitly configure the mDNS gateway using the CLI or Web UI.

Prerequisite of mDNS Gateway

Since the Cisco Catalyst 9800 series wireless controller will respond to and advertise services cached when acting as a Bonjour Gateway, it must have an SVI interface with a valid IP address on every VLAN where mDNS is allowed or used. This interface will provide the source IP address for mDNS packets that are sent from the controller acting as an mDNS Gateway.

Enable mDNS Gateway (GUI)

Enable or disable the mDNS gateway for service discovery using the GUI.

Procedure

- Step 1** Choose **Configuration > Services > mDNS**.
- Step 2** In the **Global** section, toggle the slider to enable or disable the **mDNS Gateway**.
- Step 3** From the **Transport** drop-down list, select one of these types:
- **ipv4**
 - **ipv6**
 - **both**
- Step 4** Enter an appropriate timer value in **Active-Query Timer**. The valid range is between 15 to 120 minutes. The default is 30 minutes.
- Step 5** From the **mDNS-AP Service Policy** drop-down list, select an mDNS service policy.
- Note**
Service policy is optional only if mDNS-AP is configured. If mDNS-AP is not configured, the system uses default-service-policy.
- Step 6** Click **Apply**.
-

Enable or disable mDNS Gateway (GUI)

Use this procedure to configure or change mDNS service discovery for your device using the GUI.

Procedure

- Step 1** Choose **Configuration > Services > mDNS > Global**.
- Step 2** Enable or disable the **mDNS Gateway** toggle button.
- Step 3** Select **ipv4** or **ipv6** or **both** from the **Transport** drop-down list.
- Step 4** Enter the **Active-Query Timer**.
- Step 5** Click **Apply**.
-

Enable or disable mDNS Gateway (CLI)

Enable or disable mDNS gateway to control multicast DNS service discovery for wireless devices using commands.



-
- Note**
- mDNS gateway is disabled by default globally on the controller.
 - You need both global and WLAN configurations to enable mDNS gateway.
-

Procedure

Step 1 Enable the privileged exec mode.

Example:

```
Device> enable
```

Enter your password, if prompted.

Step 2 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Enable mDNS gateway.

Example:

```
Device(config)# mdns-sd gateway
```

Step 4 Process mDNS message on a specific transport.

Example:

```
Device(config-mdns-sd)# transport {ipv4 | ipv6 | both}
```

Here,

ipv4 signifies that the IPv4 mDNS message processing is enabled. This is the default value.

ipv6 signifies that the IPv6 mDNS message processing is enabled.

both signifies that the IPv4 and IPv6 mDNS message is enabled for each network.

Step 5 Change the periodicity of mDNS multicast active query.

Example:

```
Device(config-mdns-sd)# active-query timer active-query-periodicity
```

Note

An active query is a periodic mDNS query to refresh dynamic cache.

Here, *active-query-periodicity* refers to the active query periodicity in minutes. The valid range is from 15 to 120 minutes. Active query runs with a default periodicity of 30 minutes.

Step 6 Return to the global configuration mode.

Example:

```
Device(config-mdns-sd)# exit
```

Create default service policy

Outlines the default mDNS service policy behavior and guides you in overriding it with a custom service policy if needed.

When the mDNS gateway is enabled on any of the WLANs by default, `mdns-default-service-policy` is associated with it. Default service policy consists of `default-service-list` and their details are explained in this section. You can override the default service policy with a custom service policy.

Procedure

- Step 1** Create a service-definition if the service is not listed in the preconfigured services.
 - Step 2** Create a service list for IN and OUT by using the service-definitions.
 - Step 3** Use the existing service list to create a new service. For more information, refer to *Creating Service Policy* section.
 - Step 4** Attach the `mdns-service-policy` to the profile or VLAN that needs to be enforced.
 - Step 5** To check the default-mdns-service list, use this command:
show mdns-sd default-service-list
-

Create custom service definition (GUI)

Create a service definition that specifies custom settings for mDNS through the GUI for device configuration.

Procedure

- Step 1** Choose **Configuration > Services > mDNS**.
 - Step 2** In the **Service Definition** section, click **Add**.
 - Step 3** In the **Quick Setup: Service Definition** page that is displayed, enter a name and description for the service definition.
 - Step 4** Enter a service type and click + to add the service type.
 - Step 5** Click **Apply to Device**.
-

Create custom service definition (CLI)

Add a custom service definition for mDNS, allowing you to associate user-friendly names with service types or PTR resource records using commands.

A service definition is a construct that provides an admin-friendly name to one or more mDNS service types or to a pointer (PTR) resource record name.

By default, a few built-in service definitions are already available for the admin to use.

In addition to built-in service definitions, the admin can define custom service definitions.

You can execute this command to view all service definitions (built-in and custom):

```
Device# show mdns-sd master-service-list
```

Procedure

Step 1 Enable the privileged exec mode.

Example:

```
Device> enable
```

Enter your password, if prompted.

Step 2 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Configure mDNS service definition.

Example:

```
Device(config)# mdns-sd service-definition service-definition-name
```

Note

- All the created custom service definitions are added to the primary service list.
- Primary service list comprises of a list of custom and built-in service definitions.

Step 4 Configure mDNS service type.

Example:

```
Device(config-mdns-ser-def)# service-type string
```

Step 5 Return to the global configuration mode.

Example:

```
Device(config-mdns-ser-def)# exit
```

Create service list (GUI)

Define and manage which mDNS services and message types are allowed or filtered for inbound or outbound traffic using the GUI.

Procedure

Step 1 Choose **Configuration > Services > mDNS**.

Step 2 In the **Service List** section, click **Add**.

Step 3 In the **Quick Setup: Service List** page that is displayed, enter a name for the service list.

Step 4 From the **Direction** drop-down list, select **IN** for inbound filtering or **OUT** for outbound filtering.

Step 5 From the **Available Services** drop-down list, select a service type to match the service list.

Note

To allow all services, select the **all** option.

Step 6 Click **Add Services**.

Step 7 From the **Message Type** drop-down list, select the message type to match from the following options:

- **any**: To allow all messages.
- **announcement**: To allow only service advertisements or announcements for the device.
- **query**: To allow only a query from the client for a service in the network.

Step 8 Click **Save** to add services.

Step 9 Click **Apply to Device**.

Create service list (CLI)

Create an mDNS service list in Cisco IOS XE to filter mDNS service announcements or queries for inbound or outbound traffic using commands.

mDNS service list is a collection of service definitions.

Procedure

Step 1 Enable the privileged exec mode.

Example:

```
Device> enable
```

Enter your password, if prompted.

Step 2 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Configure mDNS service list.

Example:

```
Device(config)# mdns-sd service-list service-list-name {IN | OUT}
```

- **IN**: provides inbound filtering.
- **Out**: provides outbound filtering.

Step 4 Match the service to the message type.

Example:

```
Device(config-mdns-sl-in)# match service-definition-name message-type {announcement | query}
```

Here, *service-definition-name* refers to the names of services, such as, airplay, airserver, airtunes, and so on.

Note

To add a service, the service name must be part of the primary service list.

If the mDNS service list is set to IN, you get to view this command: **match service-definition-name message-type {announcement | any | query}**.

If the mDNS service list is set to OUT, you get to view this command: **match service-definition-name**.

- Step 5** Display inbound or outbound direction list of the configured service-list to classify matching service-types for service-policy.

Example:

```
Device(config-mdns-sl-in)# show mdns-sd service-list {direction | name }
```

The list can be filtered by name or specific direction.

- Step 6** Return to the global configuration mode.

Example:

```
Device(config-mdns-sl-in)# exit
```

Create service policy (GUI)

Create a service policy and associate service lists with a specific location through the GUI.

Procedure

-
- Step 1** Choose **Configuration > Services > mDNS**.
- Step 2** In the **Service Policy** section, click **Add**.
- Step 3** In the **Quick Setup: Service Policy** page that is displayed, enter a name for the service policy.
- Step 4** From the **Service List Input** drop-down list, select one of the types.
- Step 5** From the **Service List Output** drop-down list, select one of the types.
- Step 6** From the **Location** drop-down list, select the location you want to associate with the service list.
- Step 7** Click **Apply to Device**.
-

Create service policy (CLI)

Create and apply an mDNS service policy for service filtering and control of mDNS learning and responses using commands.

mDNS service policy is used for service filtering while learning services or responding to queries.

Procedure

Step 1 Enable the privileged exec mode.

Example:

```
Device> enable
```

Enter your password, if prompted.

Step 2 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Enable mDNS service policy.

Example:

```
Device(config)# mdns-sd service-policy service-policy-name
```

Step 4 Filter mDNS service types based on LSS or site-tag.

Example:

```
Device(config-mdns-ser-pol)# location {lss | site-tag}
```

Note

In Location Specific Services (LSS) based filtering, the mDNS gateway responds with the service instances learned from the neighboring APs of the querying client AP. Service instances from other APs are filtered.

In Site tag based filtering, the mDNS gateway responds with the service instances that belong to the same site-tag as the querying client.

The mDNS gateway also responds with wired services even if location-based filtering is configured.

Step 5 Configure various service-list names for IN and OUT directions.

Example:

```
Device(config-mdns-ser-pol)# service-list service-list-name {IN | OUT}
```

Note

If an administrator creates or uses a custom service policy, the policy must have service-lists for both IN and OUT directions. Without an IN service-list, the mDNS Gateway cannot learn services. Without an OUT service-list, the mDNS Gateway cannot reply to or announce services it has learned.

Step 6 Return to the global configuration mode.

Example:

```
Device(config-mdns-ser-pol)# exit
```

Configure a local or native profile for an mDNS policy (CLI)

Ensure mDNS packets are processed based on locally defined profiles when external AAA servers do not provide a policy using commands.

When an administrator configures local authentication and authorization and does not expect to get any mDNS policy from the AAA server, the administrator can configure a local or native profile. This profile selects a mDNS policy based on user, role, or device type. When this local or native profile is mapped to the wireless profile policy, the mDNS service policy is applied to the mDNS packets that are processed on that WLAN.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the service-template or identity policy.

Example:

```
Device(config)# service-template template-name
```

Step 3 Configure the mDNS policy.

Example:

```
Device(config-service-template)# mdns-service-policy mdns-policy-name
```

Step 4 Return to the global configuration mode.

Example:

```
Device(config-service-template)# exit
```

Configure an mDNS FlexConnect profile (GUI)

Set up and apply an mDNS FlexConnect profile to define multicast DNS settings for FlexConnect devices using the GUI.

Procedure

Step 1 Choose **Configuration > Services > mDNS**.

Step 2 In the **mDNS Flex Profile** section, click **Add**.
The **Add mDNS Flex Profile** window is displayed.

Step 3 In the **Profile Name** field, enter the FlexConnect mDNS profile name.

Step 4 In the **Service Cache Update Timer** field, specify the service cache update time. The default value is 1 minute. The valid range is from 1 to 100 minutes.

Step 5 In the **Statistics Update Timer** field, specify the statistics update timer. The default value is 1 minute. The valid range is from 1 to 100 minutes.

- Step 6** In the **VLANs** field, specify the VLAN ID. You can enter multiple VLAN IDs separated by commas, or enter a range of VLAN IDs. Maximum number of VLANs allowed is 16.
- Step 7** Click **Apply to Device**.

Configure an mDNS FlexConnect Profile (CLI)

Configure an mDNS FlexConnect profile using commands to manage multicast DNS settings for FlexConnect-enabled devices. This procedure enables customization of service cache timers, statistics timers, and wired VLAN range for optimal mDNS performance.

Procedure

- Step 1** Enter the global configuration mode.

Example:

```
Device# configure terminal
```

- Step 2** Enter the mDNS FlexConnect profile mode.

Example:

```
Device(config)# mdns-sd flex-profile mdns-flex-profile-name
```

- Step 3** Configure the mDNS update service cache timer for the FlexConnect profile.

Example:

```
Device(config-mdns-flex-profile)# update-timer service-cache service-cache timer-value <1-100>
```

The default value is 1 minute. Value range is between 1 minute and 100 minutes.

- Step 4** Configure the mDNS update statistics timer for the FlexConnect profile.

Example:

```
Device(config-mdns-flex-profile)# update-timer statistics statistics timer-value <1-100>
```

The default value is 1 minute. The valid range is from 1 to 100 minutes.

- Step 5** Configure the mDNS wired VLAN range for the FlexConnect profile between 10 - 20.

Example:

```
Device(config-mdns-flex-profile)# wired-vlan-range wired-vlan-range value
```

Apply an mDNS FlexConnect profile to a wireless FlexConnect profile (GUI)

Apply an mDNS FlexConnect profile to a wireless FlexConnect profile using the GUI. This procedure enables mDNS service discovery for wireless devices connected via FlexConnect.

Procedure

- Step 1** Choose **Configuration > Tags & Profiles > Flex**.
- Step 2** Click **Add**.
The **Add Flex Profile** window is displayed.
- Step 3** Under the **General** tab, from the **mDNS Flex Profile** drop-down list, select a FlexConnect profile name from the list.
- Step 4** Click **Apply to Device**.
-

Apply an mDNS FlexConnect profile to a Wireless FlexConnect profile (CLI)

Enable mDNS features for all APs in the Wireless FlexConnect profile using commands.

Procedure

- Step 1** Enter the global configuration mode.
- Example:**
`Device# configure terminal`
- Step 2** Enter the wireless FlexConnect profile configuration mode.
- Example:**
`Device# wireless profile flex wireless-flex-profile-name`
- Step 3** Enable the mDNS features for all the APs in the profile.
- Example:**
`Device(config-wireless-flex-profile)# mdns-sd mdns-flex-profile`
-

Enable the mDNS gateway on the VLAN interface (CLI)

Enable multicast DNS (mDNS) gateway functions on a chosen VLAN interface, allowing granular service policy control for mDNS packets using commands.

This procedure configures the mDNS service policy for a specific VLAN. This allows the administrator to configure different settings to the mDNS packets on per VLAN interface basis and not on per WLAN basis.

Procedure

- Step 1** Enter the global configuration mode.
- Example:**

```
Device# configure terminal
```

Step 2 Configure a VLAN ID and enter the interface configuration mode.

Example:

```
Device(config)# interface vlan vlan-interface-number
```

Step 3 Configure the IP address for the interface.

Example:

```
Device(config-if)# ip address ip-address subnet-mask
```

Step 4 Enable mDNS configuration on a VLAN interface.

Example:

```
Device(config-if)# mdns-sd gateway
```

Step 5 Configure the service policy.

Example:

```
Device(config-if-mdns-sd)# service-policy service-policy-name
```

Note

If you do not define a specific **service-policy name**, the VLAN uses the **default-mDNS-service-policy** by default.

The system automatically creates the **default-mDNS-service-policy**, which uses the **default-mDNS-service-list** configuration to filter mDNS service announcements and queries.

Step 6 Returns to the privileged EXEC mode.

Example:

```
Device(config-if-mdns-sd)# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configure mDNS AP (CLI)

Enable Multicast DNS (mDNS) services on a specific AP and configure permitted service VLANs using commands.

In most of the deployments, the services may be available in VLANs that the APs can hear in the wired side (allowed in the switchport where the AP is directly connected: its own VLAN, or even more VLANs if switchport is a trunk).

This procedure shows how to configure mDNS AP:

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the mDNS gateway.

Example:

```
Device(config)# mdns-sd gateway
```

Step 3 Enable mDNS on the AP, and configure a VLAN for the mDNS AP.

Example:

```
Device# ap name ap-name mdns-ap enable vlan vlan-id
```

Step 4 Add a VLAN to the mDNS AP.

Example:

```
Device# ap name ap-name mdns-ap vlan add vlan-id
```

The *vlan-id* ranges from 1 to 4096.

Step 5 Delete a VLAN from the mDNS AP.

Example:

```
Device# ap name ap-name mdns-ap vlan del vlan-id
```

Step 6 Disable the mDNS AP.

Example:

```
Device# ap name ap-name mdns-ap disable
```

Step 7 Return to the privileged EXEC mode.

Example:

```
Device# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Note

You can configure a maximum of 10 VLANs per AP.

Enable mDNS Gateway on the RLAN interface (CLI)

Enable mDNS gateway to allow devices connected to a Remote LAN (RLAN) interface to discover services across VLANs using commands.

By configuring the mDNS gateway mode on the RLAN interface, you can configure the mDNS service policy for a specific RLAN.

Procedure

Step 1 Enter the global configuration mode and configure a remote LAN profile.

Example:

```
Device# configure terminal
```

```
Device(config)# ap remote-lan profile-name remote-lan-profile-name rlan-id
```

Example:

- *remote-lan-profile*: Remote LAN profile name. The range is from 1 to 32 alphanumeric characters.
- *rlan-id*: Remote LAN identifier. The range is from 1 to 128.

Note

You can create a maximum of 128 RLANs. Also, you cannot use the *rlan-id* of an existing RLAN while creating another RLAN.

Step 2 Enable mDNS configuration on an RLAN interface and restart the RLAN profile.

Example:

```
mdns-sd-interface
```

```
Device(config-remote-lan)# mdns-sd-interface {gateway | drop}
```

```
Device(config-remote-lan)# no shutdown
```

Step 3 Exit the remote LAN configuration mode and configure the RLAN policy profile and enter the wireless policy configuration mode.

Example:

```
Device(config-remote-lan)# exit
```

```
Device(config)# ap remote-lan-policy policy-name profile name
```

Step 4 Enable an mDNS service policy and configure the RLAN for central switching.

Example:

```
Device(config-remote-lan-policy)# mdns-sd service-policy service-policy-name
```

```
Device(config-remote-lan-policy)# central switching
```

Step 5 Configure the central DHCP for centrally switched clients and assign the profile policy to a VLAN.

Example:

```
Device(config-remote-lan-policy)# central dhcp
```

```
Device(config-remote-lan-policy)# vlan vlan-name
```

Step 6 Restart the RLAN profile and configure a policy tag.

Example:

```
Device(config-remote-lan-policy)# no shutdown
```

```
Device(config)# wireless tag policy policy-tag-name
```

Step 7 Map the RLAN policy profile to the RLAN profile.

Example:

```
Device(config-policy-tag)# remote-lan remote-lan-profile-name policy rlan-policy-profile-name
port-id port-id
```

- *remote-lan-profile-name* : Name of the RLAN profile.
- *rlan-policy-profile-name*: Name of the policy profile.
- *port-id*: LAN port number on the AP. The range is from 1 to 4.

Step 8 Return to the global configuration mode and configure the AP and enter the AP tag configuration mode.

Example:

```
Device(config-policy-tag)# exit
Device (config)# ap mac-address
```

Note

Use the Ethernet MAC address.

Step 9 Map a policy tag to the AP and return to the privileged EXEC mode.

Example:

```
Device (config-ap-tag)# policy-tag policy-tag-name
Device(config-guest-lan)# end
```

Enable mDNS Gateway on guest LAN interface (CLI)

Enable the mDNS gateway on a guest LAN interface to support discovery and service policies for wired and anchor controllers using commands.

By configuring the mDNS gateway mode on a Guest LAN interface, you can configure the mDNS service policy for a specific Guest LAN interface.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure guest LAN profile with a wired VLAN.

Example:

```
Device(config)# guest-lan profile-name guest_lan_profile_name num wired-vlan wired_vlan_num
```

Note

Configures the wired VLAN only for the Guest Foreign controller.

- *num*: Guest LAN identifier. The valid range is from 1 to 5.
- *wired_vlan_num*: Wired VLAN number. The valid range is from 1 to 4094.

Step 3 Configure the guest LAN profile without a VLAN for the Guest Anchor controller.

Example:

```
Device(config)# guest-lan profile-name guest_lan_profile_name
```

Step 4 Configure the mDNS gateway for a Guest LAN.

Example:

```
Device(config-guest-lan)# mdns-sd interface {gateway | drop}
```

Note

You need to enable mDNS gateway globally for the Guest LAN to work.

Step 5 Return to the privileged EXEC mode.

Example:

```
Device(config-guest-lan)# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Associate mDNS service policy with wireless profile policy (GUI)

Apply an mDNS service policy to a wireless profile policy so that multicast DNS behavior aligns with desired network access and discovery parameters using the GUI.

Procedure

Step 1 Choose **Configuration > Tags & Profiles > Policy**.

Step 2 Click the **policy profile** name.

Step 3 In the **Advanced** tab, select the mDNS service policy from the **mDNS Service Policy** drop-down list.

Step 4 Click **Update & Apply to Device**.

Associate mDNS service policy with wireless profile policy (CLI)

Enable customized mDNS service filtering and announcements within a wireless profile policy using commands.



Note You must globally configure the mDNS service policy before associating it with the wireless profile policy.

A default mDNS service policy is already attached when the wireless profile policy is created. You can use the following commands to override the default mDNS service policy with any of your service policy:

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configures the wireless profile policy.

Example:

```
Device(config)# wireless profile policy profile-policy
```

Here, *profile-policy* refers to the name of the WLAN policy profile.

Step 3 Associate an mDNS service policy with the wireless profile policy.

Example:

```
Device(config-wireless-policy)# mdns-sd service-policy custom-mdns-service-policy
```

The default mDNS service policy name is **default-mdns-service-policy**.

Note

The **default-mdns-profile-policy** uses **default-mdns-service-list** configuration to filter mDNS service announcements and queries.

In a wireless network, the mDNS packets are consumed by the mDNS gateway, so clients or devices are unable to learn about these services. By sharing the service with the device and simplifying configuration for the administrator, a list of standard service types is shared by default on the wireless network. This list of standard service types is referred to as the default service policy, which comprises a set of service types.

The table covers a sample service list in the default service policy.

Table 1: Default Name and mDNS Service Type

Default Name	mDNS Service Type
Apple HomeSharing	_home-sharing._tcp.local
Printer-IPPS	_ipps._tcp.local
Google-chromecast	_googlecast._tcp.local

Note

- Location would be disabled on mDNS default service policy.
- You cannot change the contents of the mDNS default service policy. However, you can create separate mDNS service policies and associate them under the wireless policy profile.

Step 4 Return to the global configuration mode.

Example:

```
Device(config-wireless-policy)# exit
```

Enable or disable mDNS Gateway for WLAN (GUI)

Enable or disable the mDNS Gateway feature for a specific WLAN to optimize device discovery and network performance using the GUI.

Procedure

Step 1 Choose **Configuration > Tags & Profiles > WLANs**.

Step 2 Click on the WLAN.

- Step 3** In the **Advanced** tab, select the mode in **mDNS Mode** drop-down list.
- Step 4** Click **Update & Apply to Device**.

Enable or disable mDNS Gateway for WLAN (CLI)

Control mDNS Gateway functionality for a wireless LAN using commands.



Note Bridging is the default behaviour. This means that the mDNS packets are always bridged.

Procedure

- Step 1** Enter the global configuration mode.

Example:

```
Device# configure terminal
```

- Step 2** Specify the WLAN name and ID.

Example:

```
Device(config)# wlan profile-name wlan-id ssid-name
```

- *profile-name* is the WLAN name, and it can contain up to 32 alphanumeric characters.
- *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 512 .
- *ssid-name* is the SSID which can contain 32 alphanumeric characters.

Note

Global configuration must be in place for mDNS Gateway to work.

- Step 3** Enable or disable mDNS Gateway and bridge functions on WLAN.

Example:

```
Device(config-wlan)# mdns-sd interface {gateway | drop}
```

- Step 4** Return to the global configuration mode.

Example:

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

- Step 5** Verify the status of mDNS on WLAN.

Example:

```
Device# show wlan name wlan-name show wlan all
```

- Step 6** Verify the service policy configured in WLAN.

Example:

```
Device# show wireless profile policy
```

mDNS Gateway with guest anchor support and mDNS bridging

- When mDNS Gateway is enabled on both the Anchor and Foreign controllers, the mDNS gateway supports guest anchor deployments. In this mode, clients on a guest LAN or WLAN with guest anchor enabled receive responses with any services or cache from the export Foreign controller. All advertisements received on the guest LAN or WLAN by the export Foreign are learned on the export Foreign itself. All queries received on the guest LAN or WLAN are answered by the export Foreign itself.
- When mDNS Gateway is enabled on the Anchor and disabled on the Foreign controller (Bridging Mode), the mDNS gateway supports guest anchor deployments. In this scenario, clients on the guest LAN or WLAN with guest anchor enabled receive any services or cache from the export Anchor, even if the clients are connected to the Foreign. All advertisements received on the guest LAN or WLAN by the export Foreign are forwarded to the Anchor, and the cache is stored on the Anchor. All queries received on the guest LAN or WLAN are answered by the export Anchor itself.



Note Comment by CISCO\ashijadh: based on PRRQ comments

- You must configure the guest LAN to a wireless profile policy that is configured with the required mDNS service policy.
 - To configure a non-guest LAN mDNS gateway, see the mDNS Gateway chapter.
-

Configure mDNS Gateway on guest anchor (CLI)

Configures the mDNS Gateway functionality on a guest anchor device using commands.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the guest LAN profile with a wired VLAN.

Example:

```
Device(config)# guest-lan profile-name guest-lan-profile-name guest-lan-id
```

Step 3 Enable mDNS Gateway on the guest LAN.

Example:

```
Device(config-guest-lan)# mdns-sd gateway
```

Configure mDNS Gateway on guest foreign (Guest LAN) (CLI)

Enable mDNS Gateway functionality for users and devices connected to the Guest LAN via a wired VLAN using commands.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure guest LAN profile with a wired VLAN.

Example:

```
Device(config)# guest-lan profile-name guest-lan-profile-name guest-lan-id wired-vlan vlan-id
```

Note

Configure the wired VLAN only for the Guest Foreign controller.

Step 3 Enable mDNS Gateway on the guest LAN.

Example:

```
Device(config-guest-lan)# mdns-sd gateway
```

Step 4 Return to the global configuration mode.

Example:

```
Device(config-wireless-policy)# exit
```

Configure mDNS Gateway on guest anchor (CLI)

Enable mDNS Gateway functionality on a guest WLAN profile anchored to a VLAN, allowing multicast DNS service discovery for guest users using commands.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the guest WLAN profile with a wired VLAN.

Example:

```
Device(config)# guest-wlan profile-name guest-lan-profile-name guest-wlan-id
```

Step 3 Enable mDNS Gateway on the guest WLAN.

Example:

```
Device(config-guest-wlan)# mdns-sd gateway
```

Configure mDNS Gateway on guest foreign (Guest WLAN) (CLI)

Enable service discovery for devices on a guest wireless LAN (WLAN) and ensure multicast DNS (mDNS) works for clients connected to a guest wired VLAN on the foreign controller using commands.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure guest WLAN profile with a wired VLAN.

Example:

```
Device(config)# guest-wlan profile-name guest-lan-profile-name guest-wlan-id wired-vlan vlan-id
```

Note

Configure the wired VLAN only for the Guest Foreign controller.

Step 3 Enable mDNS Gateway on the guest WLAN.

Example:

```
Device(config-guest-wlan)# mdns-sd gateway
```

Step 4 Return to the global configuration mode.

Example:

```
Device(config-wireless-policy)# exit
```

Verify mDNS Gateway configurations

To verify the mDNS summary, use this command:

```
Device# show mdns-sd summary
```

```
mDNS Gateway: Enabled
Active Query: Enabled
  Periodicity (in minutes): 30
Transport Type: IPv4
```

To verify the mDNS cache, use this command:

```
Device# show mdns-sd cache
```

```
----- PTR Records
-----
RECORD-NAME                               TTL      WLAN    CLIENT-MAC    RR-RECORD-DATA
```

Verify mDNS Gateway configurations

```
-----
_airplay._tcp.local          4500    30    07c5.a4f2.dc01  CUST1._airplay._tcp.local
_ipp._tcp.local             4500    30    04c5.a4f2.dc01  CUST3._ipp._tcp.local2
_ipp._tcp.local             4500    15    04c5.a4f2.dc01  CUST3._ipp._tcp.local4
_ipp._tcp.local             4500    10    04c5.a4f2.dc01  CUST3._ipp._tcp.local6
_veer_custom._tcp.local     4500    10    05c5.a4f2.dc01
CUST2._veer_custom._tcp.local8
```

To verify the mDNS cache from wired service provider, use this command:

```
Device# show mdns-sd cache wired
```

```
----- PTR Records
-----
RECORD-NAME                TTL      VLAN      CLIENT-MAC      RR-RECORD-DATA
-----
_airplay._tcp.local        4500     16        0866.98ec.97af
wiredapple._airplay._tcp.local
_raop._tcp.local          4500     16        0866.98ec.97af
086698EC97AF@wiredapple._raop._tcp.local
----- SRV Records
-----
RECORD-NAME                TTL      VLAN      CLIENT-MAC      RR-RECORD-DATA
-----
wiredapple._airplay._tcp.local  4500     16        0866.98ec.97af  0 0 7000
wiredapple.local
086698EC97AF@wiredapple._raop._tcp.local  4500     16        0866.98ec.97af  0 0 7000
wiredapple.local
----- A/AAAA Records
-----
RECORD-NAME                TTL      VLAN      CLIENT-MAC      RR-RECORD-DATA
-----
wiredapple.local          4500     16        0866.98ec.97af
2001:8:16:16:e5:c446:3218:7437
----- TXT Records
-----
RECORD-NAME                TTL      VLAN      CLIENT-MAC      RR-RECORD-DATA
-----
wiredapple._airplay._tcp.local  4500     16        0866.98ec.97af
[343]'acl=0''deviceid=08:66:98:EC:97:AF''features=
086698EC97AF@wiredapple._raop._tcp.local  4500     16        0866.98ec.97af
[193]'cn=0,1,2,3''da=true''et=0,3,5''ft=0x5A7FFF7
```

To verify the mdns-sd type PTR, use this command:

```
Device# show mdns-sd cache type {PTR | SRV | A-AAA | TXT}
```

```
RECORD-NAME                TTL      WLAN      CLIENT-MAC
RR-Record-Data
-----
_custom1._tcp.local        4500     2         c869.cda8.77d6
service_tl_custom1._tcp.local
_custom1._tcp.local        4500     2         c869.cda8.77d6
vk11._custom1._tcp.local
_ipp._tcp.local            4500     2         c869.cda8.77d6
service-4._ipp._tcp.local
```

To verify the mdns-sd cache for a client MAC, use this command:

```
Device# show mdns-sd cache {ap-mac <ap-mac> | client-mac <client-mac>
```

```

| glan-id <glan-id> | mdns-ap <mac-address> | rlan-id <rlan-id>
| wlan-id <wlan-id> | wired)
RECORD-NAME          TTL      WLAN      CLIENT-MAC
RR-Record-Data
-----
_custom1._tcp.local          4500     2          c869.cda8.77d6
service_t1._custom1._tcp.local
_custom1._tcp.local          4500     2          c869.cda8.77d6
vk11._custom1._tcp.local
_ipp._tcp.local              4500     2          c869.cda8.77d6
service-4._ipp._tcp.local
----- SRV Records -----
RECORD-NAME          TTL      WLAN      CLIENT-MAC
RR-Record-Data
-----
service-4._ipp._tcp.local          4500     2          c869.cda8.77d6   0 0 1212
mDNS-Client1s-275.local
vk11._custom1._tcp.local          4500     2          c869.cda8.77d6   0 0 987
mDNS-Client1s-275.local
service_t1._custom1._tcp.local     4500     2          c869.cda8.77d6   0 0 197
mDNS-Client1s-275.local
----- A/AAAA Records -----
RECORD-NAME          TTL      WLAN      CLIENT-MAC
RR-Record-Data
-----
mDNS-Client1s-275.local          4500     2          c869.cda8.77d6   120.1.1.33
----- TXT Records -----
RECORD-NAME          TTL      WLAN      CLIENT-MAC
RR-Record-Data
-----
service-4._ipp._tcp.local          4500     2          c869.cda8.77d6   'Client1'
vk11._custom1._tcp.local          4500     2          c869.cda8.77d6
'stxtvers=11'
service_t1._custom1._tcp.local     4500     2          c869.cda8.77d6
'stxtvers=12'

```

To verify the mdns-sd cache with respect to the RLAN ID, use this command:

```
Device# show mdns-sd cache rlan-id 1 detail
```

```

Name: _printer._tcp.local
Type: PTR
TTL: 4500
RLAN: 1
RLAN Name: rlan_test_1
VLAN: 141
Client MAC: 000e.c688.3942
AP Ethernet MAC: 0042.5ab6.0ef0
Remaining-Time: 4485
Site-Tag: default-site-tag
mDNS Service Policy: mdnsTV6
Overriding mDNS Service Policy: NO
UPN-Status: Disabled
Rdata: printer._printer._tcp.local

```

```
Name: lab-47-187.local
Type: A/AAAA
TTL: 4500
RLAN: 1
RLAN Name: rlan_test_1
VLAN: 141
Client MAC: 000e.c688.3942
AP Ethernet MAC: 0042.5ab6.0ef0
Remaining-Time: 4485
Site-Tag: default-site-tag
mDNS Service Policy: mdnsTV6
Overriding mDNS Service Policy: NO
UPN-Status: Disabled
Rdata: 10.15.141.124
```

To verify the mdns-sd cache with respect to mDNS-AP, use this command:

```
Device# show mdns-sd cache mdns-ap 706b.b97d.b060 detail
```

```
Name: _printer._tcp.local
Type: PTR
TTL: 4500
VLAN: 145
Client MAC: 0050.b626.5bfa
mDNS AP Radio MAC: 706b.b97d.b060
mDNS AP Ethernet MAC: 706b.b97c.5208
Remaining-Time: 4480
mDNS Service Policy: mdnsTV
Rdata: printer._printer._tcp.local
Name: Client-46-153.local
Type: A/AAAA
TTL: 4500
VLAN: 145
Client MAC: 0050.b626.5bfa
mDNS AP Radio MAC: 706b.b97d.b060
mDNS AP Ethernet MAC: 706b.b97c.5208
Remaining-Time: 4480
mDNS Service Policy: mdnsTV
Rdata: 10.15.145.103
```

To verify the mdns-sd cache in detail, use this command:

```
Device# show mdns-sd cache detail
```

```
Name: _custom1._tcp.local
Type: PTR
TTL: 4500
WLAN: 2
WLAN Name: mdns120
VLAN: 120
Client MAC: c869.cda8.77d6
AP Ethernet MAC: 7069.5ab8.33d0
Expiry-Time: 09/09/18 21:50:47
Site-Tag: default-site-tag
Rdata: service_t1._custom1._tcp.local
```

To verify the mdns-sd cache statistics, use this command:

```
Device# show mdns-sd cache statistics
```

```
mDNS Cache Stats
Total number of Services: 4191
```

To verify the mdns-sd statistics, use this command:

```
Device# show mdns-sd statistics

-----
Consolidated mDNS Packet Statistics
-----
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 61045
  IPv4 sent: 30790
    IPv4 advertisements sent: 234
    IPv4 queries sent: 30556
  IPv6 sent: 30255
    IPv6 advertisements sent: 17
    IPv6 queries sent: 30238
  Multicast sent: 57558
    IPv4 sent: 28938
    IPv6 sent: 28620
mDNS packets received: 72796
  advertisements received: 13604
  queries received: 59192
  IPv4 received: 40600
    IPv4 advertisements received: 6542
    IPv4 queries received: 34058
  IPv6 received: 32196
    IPv6 advertisements received: 7062
    IPv6 queries received: 25134
mDNS packets dropped: 87
-----
Wired mDNS Packet Statistics
-----
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 61033
  IPv4 sent: 30778
    IPv4 advertisements sent: 222
    IPv4 queries sent: 30556
  IPv6 sent: 30255
    IPv6 advertisements sent: 17
    IPv6 queries sent: 30238
  Multicast sent: 57558
    IPv4 sent: 28938
    IPv6 sent: 28620
mDNS packets received: 52623
  advertisements received: 1247
  queries received: 51376
  IPv4 received: 32276
    IPv4 advertisements received: 727
    IPv4 queries received: 31549
  IPv6 received: 20347
    IPv6 advertisements received: 520
    IPv6 queries received: 19827
mDNS packets dropped: 63
-----
mDNS Packet Statistics, for WLAN: 2
-----
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 12
  IPv4 sent: 12
    IPv4 advertisements sent: 12
    IPv4 queries sent: 0
  IPv6 sent: 0
    IPv6 advertisements sent: 0
    IPv6 queries sent: 0
  Multicast sent: 0
```

```

    IPv4 sent: 0
    IPv6 sent: 0
mDNS packets received: 20173
  advertisements received: 12357
  queries received: 7816
  IPv4 received: 8324
    IPv4 advertisements received: 5815
    IPv4 queries received: 2509
  IPv6 received: 11849
    IPv6 advertisements received: 6542
    IPv6 queries received: 5307
mDNS packets dropped: 24

```

To verify the default service list details, use this command:

```
Device# show mdns-sd default-service-list
```

```

-----
                mDNS Default Service List
-----
Service Definition: airplay
Service Names: _airplay._tcp.local
Service Definition: airtunes
Service Names: _raop._tcp.local
Service Definition: homesharing
Service Names: _home-sharing._tcp.local
Service Definition: printer-ipp
Service Names: _ipp._tcp.local
Service Definition: printer-lpd
Service Names: _printer._tcp.local
Service Definition: printer-ipps
Service Names: _ipps._tcp.local
Service Definition: printer-socket
Service Names: _pdl-datastream._tcp.local
Service Definition: google-chromecast
Service Names: _googlecast._tcp.local
Service Definition: itune-wireless-devicesharing2
Service Names: _apple-mobdev2._tcp.local

```

To verify the primary service list details, use this command:

```
Device# show mdns-sd master-service-list
```

```

-----
                mDNS Master Service List
-----
Service Definition: fax
Service Names: _fax-ipp._tcp.local
Service Definition: roku
Service Names: _rsp._tcp.local
Service Definition: airplay
Service Names: _airplay._tcp.local
Service Definition: scanner
Service Names: _scanner._tcp.local
Service Definition: spotify
Service Names: _spotify-connect._tcp.local
Service Definition: airtunes
Service Names: _raop._tcp.local
Service Definition: airserver
Service Names: _airplay._tcp.local
                _airserver._tcp.local
.
.
.

```

```
Service Definition: itune-wireless-devicesharing2
Service Names: _apple-mobdev2._tcp.local
```

To verify the mdns-sd service statistics on the controller, use this command:

```
Device# show mdns-sd service statistics
```

Service Name	Service Count
_atc._tcp.local	137
_hap._tcp.local	149
_ipp._tcp.local	149
_rfb._tcp.local	141
_smb._tcp.local	133
_ssh._tcp.local	142
_daap._tcp.local	149
_dpap._tcp.local	149
_eppc._tcp.local	138
_adisk._tcp.local	149

To verify the mDNS-AP configured on the controller and VLAN(s) associated with it, use this command:

```
Device# show mdns-sd ap
```

```
Number of mDNS APs..... 1
AP Name   Ethernet MAC   Number of Vlans   Vlanidentifiers
-----
AP3600-1  7069.5ab8.33d0   1                 300
```

Further Debug

To debug mDNS further, use this procedure:

1. Run this command at the controller:

```
set platform software trace wncd <0-7> chassis active R0 mdns debug
```

2. Reproduce the issue.

3. Run this command to gather the traces enabled:

```
show wireless loadbalance ap affinity wncd 0
AP MAC   Discovery Timestamp   Join Timestamp           Tag   Vlanidentifiers
-----
0cd0.f894.0600   06/30/21 12:39:48   06/30/21 12:40:021   default-site-tag   300
```

Location based service filtering

Prerequisite for location-based service filtering

- You need to create the Service Definition and Service Policy.
- For more information, see [Creating Custom Service Definition](#) section and [Creating Service Policy](#) section.

Configure mDNS location-based filtering using SSID (CLI)

Limit mDNS service announcements to only those learned on the configured SSID, enhancing security and service relevance using commands.

When a service policy is configured with the SSID as the location name, the response to the query will be the services that were learnt on that SSID.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the service policy.

Example:

```
Device(config)# mdns-sd service-policy service-policy-name
```

Step 3 Configure location-based filtering using SSID.

Example:

```
Device(config-mdns-ser-pol)# location ssid
```

Step 4 Return to the privileged EXEC mode.

Example:

```
Device(config-mdns-ser-pol)# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configure mDNS location-based filtering using AP name (CLI)

Limit mDNS query responses to only services discovered on a specified AP, improving service localization and security using commands.

When a service policy is configured with the AP name as the location, the response to the query will be the services that were learnt on that AP.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the service policy.

Example:

```
Device(config)# mdns-sd service-policy service-policy-name
```

Step 3 Configure location-based filtering using an AP name.

Example:

```
Device(config-mdns-ser-pol)# location ap-name
```

Step 4 Return to the privileged EXEC mode.

Example:

```
Device(config-mdns-ser-pol)# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configure mDNS location-based filtering using AP location (CLI)

Configure mDNS location-based filtering using the AP location and these commands.

When a service policy is configured with location as the AP-location, the response to the query will be the services that were learnt on all the APs using the same AP "location" name (not to be confused with "site-tag").

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the service policy.

Example:

```
Device(config)# mdns-sd service-policy service-policy-name
```

Step 3 Configure location-based filtering using the AP location.

Example:

```
Device(config-mdns-ser-pol)# location ap-location
```

Step 4 Return to the privileged EXEC mode.

Example:

```
Device(config-mdns-ser-pol)# end
```

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configure mDNS location-based filtering using regular expression (CLI)

Enable filtering of mDNS services using location-based regular expressions for targeted service policy assignment using commands.

- If a service policy uses a location specified as a regular expression matching the AP name, the query returns services learned on all APs whose names match.
- If a service policy uses a location specified as a regular expression matching the AP location, the query returns services learned on all APs whose locations match.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the service policy.

Example:

```
Device(config)# mdns-sd service-policy service-policy-name
```

Step 3 Configure location-based filtering using regular expression.

Example:

```
Device(config-mdns-ser-pol)# location regex {ap-location regular-expression | ap-name  
regular-expression}
```

Step 4 Return to the privileged EXEC mode.

Example:

```
Device(config-mdns-ser-pol)# end
```

Note

To filter the services for which AP names have the specific keyword such as *AP-2FLR-SJC-123*, you can use the regex AP name as *AP-2FLR-* to match the services that are learnt from the set of APs.

Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.
