



Native Profiling

- [Information About Native Profiling, on page 1](#)
- [Creating a Class Map \(GUI\), on page 2](#)
- [Creating a Class Map \(CLI\), on page 3](#)
- [Creating a Service Template \(GUI\), on page 5](#)
- [Creating a Service Template \(CLI\), on page 6](#)
- [Creating a Parameter Map, on page 7](#)
- [Creating a Policy Map \(GUI\), on page 7](#)
- [Creating a Policy Map \(CLI\), on page 8](#)
- [Configuring Native Profiling in Local Mode, on page 10](#)
- [Verifying Native Profile Configuration, on page 10](#)

Information About Native Profiling

You can profile devices based on HTTP and DHCP to identify the end devices on the network. You can configure device-based policies and enforce these policies per user or per device policy on the network.

Policies allow profiling of mobile devices and basic onboarding of the profiled devices to a specific VLAN. They also assign ACL and QoS or configure session timeouts.

The policies are defined based on the following attributes:

- User group or user role
- Device type such as Windows clients, smartphones, tablets, and so on
- Service Set Identifier (SSID)
- Location, based on the access point group that the end point is connected to
- Time of the day
- Extensible Authentication Protocol (EAP) type, to check what EAP method that the client is getting connected to

When a wireless client joins an access point, certain QoS policies get enforced on the access point. One such feature is the native profiling for both upstream and downstream traffic at AP. The native profiling feature when clubbed with AAA override supports specific set of policies based on the time of day and day of week. The AAA override then applies these policies coming from a RADIUS server to the access point.

Let's consider a use case of time of the day in conjunction with user role. Usually, the user role is used as an extra matching criteria along with the time of day. You can club the time of day usage with any matching criteria to get the desired result. The matching will be performed when the client joins the controller .

You can configure policies as two separate components:

- Defining policy attributes as service templates that are specific to clients joining the network and applying policy match criteria
- Applying match criteria to the policy.



Note Before proceeding with the native profile configuration, ensure that HTTP Profiling and DHCP Profiling are enabled.



Note Native profiling is not supported with FlexConnect Local Authentication and Local Switching. Hence, do not configure **no central switching**, **no central authentication**, and **subscriber-policy-name name** commands together. ISSU will fail for this type of configuration. Ensure that you remove the configuration before attempting ISSU.

To configure Native Profiling, use one of the following procedures:

- Create a service template
- Create a class map



Note You can apply a service template using either a class map or parameter map.

- Create a parameter-map and associate the service template to parameter-map
 - Create a policy map
 1. If class-map has to be used: Associate the class-map to the policy-map and associate the service-template to the class-map.
 2. If parameter-map has to be used: Associate the parameter-map to the policy-map
 - Associate the policy-map to the policy profile.

Creating a Class Map (GUI)

Procedure

Step 1 Click **Configuration > Services > QoS**.

Step 2 In the **QoS – Policy** area, click **Add** to create a new QoS Policy or click the one you want to edit.

- Step 3** Add **Add Class Map** and enter the details.
- Step 4** Click **Save**.
- Step 5** Click **Update and Apply to Device**.

Creating a Class Map (CLI)



Note Configuration of class maps via CLI offer more options and can be more granular than GUI.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | class-map type control subscriber match-any class-map-name Example: Device(config)# class-map type control subscriber match-any cls_user | Specifies the class map type and name. |
| Step 3 | match username username Example: Device(config-filter-control-classmap)# match username ciscoise | Specifies the class map attribute filter criteria. |
| Step 4 | class-map type control subscriber match-any class-map-name Example: Device(config)# class-map type control subscriber match-any cls_userrole | Specifies the class map type and name. |
| Step 5 | match user-role user-role Example: Device(config-filter-control-classmap)# match user-role engineer | Specifies the class map attribute filter criteria. |
| Step 6 | class-map type control subscriber match-any class-map-name Example: Device(config)# class-map type control subscriber match-any cls_oui | Specifies the class map type and name. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 7 | match oui <i>oui-address</i> Example: <pre>Device(config-filter-control-classmap)# match oui 48.f8.b3</pre> | Specifies the class map attribute filter criteria. |
| Step 8 | class-map type control subscriber match-any <i>class-map-name</i> Example: <pre>Device(config)# class-map type control subscriber match-any cls_mac</pre> | Specifies the class map type and name. |
| Step 9 | match mac-address <i>mac-address</i> Example: <pre>Device(config-filter-control-classmap)# match mac-address 0040.96b9.4a0d</pre> | Specifies the class map attribute filter criteria. |
| Step 10 | class-map type control subscriber match-any <i>class-map-name</i> Example: <pre>Device(config)# class-map type control subscriber match-any cls_devtype</pre> | Specifies the class map type and name. |
| Step 11 | match device-type <i>device-type</i> Example: <pre>Device(config-filter-control-classmap)# match device-type windows</pre> | Specifies the class map attribute filter criteria. |
| Step 12 | class-map type control subscriber match-all <i>class-map-name</i> Example: <pre>Device(config)# class-map type control subscriber match-all match_tod</pre> | Specifies the class map type and name. |
| Step 13 | match join-time-of-day <i>start-time end-time</i> Example: <pre>Device(config-filter-control-classmap)# match join-time-of-day 10:30 12:30</pre> | <p>Specifies a match to the time of day.</p> <p>Here, join time is considered for matching. For example, if the match filter is set from 11:00 am to 2:00 pm, a device joining at 10:59 am is not considered, even if it acquires credentials after 11:00 am.</p> <p>Here,</p> <p><i>start-time</i> and <i>end-time</i> specifies the 24-hour format.</p> <p>Use the show class-map type control subscriber name <i>name</i> command to verify the configuration.</p> |

| | Command or Action | Purpose |
|----------------|--|---|
| | | Note You should also disable AAA override for this command to work. |
| Step 14 | match day <i>day-of-week</i> Example: Device(config-filter-control-classmap)# match day Monday | Matches day of the week. Use the show class-map type control subscriber name <i>name</i> command to verify the configuration. |
| Step 15 | class-map type control subscriber match-all <i>class-map-name</i> Example: Device(config)# class-map type control subscriber match-all match_eap | Specifies the class map type and filter as EAP. |
| Step 16 | match eap-type <i>eap-type</i> Example: Device(config-filter-control-classmap)# match eap-type peap | Specifies the policy match with EAP type. Use the show class-map type control subscriber name <i>name</i> command to verify the configuration. |
| Step 17 | class-map type control subscriber match-all <i>class-map-name</i> Example: Device(config)# class-map type control subscriber match-all match_device | Specifies the class map type and filter as device. |
| Step 18 | match device-type <i>device-name</i> Example: Device(config-filter-control-classmap)# match device-type android | Matches name using the device type. Type a question mark (?) after the device type and select the device from the list. Note You should enable the device classifier for the device list to be populated. |

Creating a Service Template (GUI)

Procedure

-
- Step 1** Choose **Configuration > Security > Local Policy**.
- Step 2** On the **Local Policy** page, **Service Template** tab, click **ADD**.
- Step 3** In the **Create Service Template** window, enter the following parameters:
- **Service Template Name:** Enter a name for the template.
 - **VLAN ID:** Enter the VLAN ID for the template. Valid range is between 1 and 4094.

- **Session Timeout (secs):** Sets the timeout duration for the template. Valid range is between 1 and 65535.
- **Access Control List:** Choose the Access Control List from the drop-down list.
- **Ingress QoS:** Choose the input QoS policy for the client from the drop-down list
- **Egress QoS:** Choose the output QoS policy for the client from the drop-down list.

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

Creating a Service Template (CLI)

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | service-template <i>service-template-name</i> Example: Device(config)# service-template svcl | Enters service template configuration mode. |
| Step 3 | vnid <i>vnid</i> Example: Device(config-service-template)# vnid test | Specifies the VXLAN network identifier (VNID). Use the show service-template <i>service-template-name</i> command to verify the configuration. |
| Step 4 | access-group <i>access-list-name</i> Example: Device(config-service-template)# access-group acl-auto | Specifies the access list to be applied. |
| Step 5 | vlan <i>vlan-id</i> Example: Device(config-service-template)# vlan 10 | Specifies VLAN ID. Valid range is from 1-4094. |
| Step 6 | absolute-timer <i>timer</i> Example: Device(config-service-template)# absolute-timer 1000 | Specifies session timeout value for a service template. Valid range is from 1-65535. |
| Step 7 | service-policy qos input <i>qos-policy</i> Example: | Configures an input QoS policy for the client. |

| | Command or Action | Purpose |
|---------------|---|---|
| | Device(config-service-template)# service-policy qos input in_qos | |
| Step 8 | service-policy qos output qos-policy Example: Device(config-service-template)# service-policy qos output out_qos | Configures an output QoS policy for the client. |

Creating a Parameter Map

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | parameter-map type subscriber attribute-to-service parameter-map-name Example: Device(config)# parameter-map type subscriber attribute-to-service param | Specifies the parameter map type and name. |
| Step 3 | map-indexmap device-type eqfilter-name Example: Device(config-parameter-map-filter)# 1 map device-type eq "windows" mac-address eq 3c77.e602.2f91 username eq "cisco" | Specifies the parameter map attribute filter criteria. Multiple filters are used in the example provided here. |
| Step 4 | map-indexservice-template service-template-name precedence precedence-num Example: Device(config-parameter-map-filter-submode)# 1 service-template svcl precedence 150 | Specifies the service template and its precedence. |

Creating a Policy Map (GUI)

Procedure

-
- Step 1** Choose **Configuration > Security > Local Policy > Policy Map** tab..
- Step 2** Enter a name for the Policy Map in the **Policy Map Name** text field.

- Step 3** Click **Add**
- Step 4** Choose the service template from the **Service Template** drop-down list.
- Step 5** For the following parameters select the type of filter from the drop-down list and enter the required match criteria
- Device Type
 - User Role
 - User Name
 - OUI
 - MAC Address
- Step 6** Click **Add Criteria**
- Step 7** Click **Update & Apply to Device**.
-

Creating a Policy Map (CLI)

Before you begin

Before removing a policy map or parameter map, you should remove it from the target or shut down the WLAN profile or delete the session.

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | policy-map type control subscriber <i>policy-map-name</i> Example: Device(config)# policy-map type control subscriber polmap5 | Specifies the policy map type. |
| Step 3 | event identity-update match-all Example: Device(config-event-control-policymap)# event identity-update match-all | Specifies the match criteria to the policy map. |
| Step 4 | You can apply a service template using either a class map or a parameter map, as shown here. • <i>class-num</i> class <i>class-map-name</i> do-until-failure | Configures the local profiling policy class map number and specifies how to perform the action or activates the service template or maps an identity-update attribute to an auto-configured template. |

| | Command or Action | Purpose |
|---------------|--|---|
| | <ul style="list-style-type: none"> • <i>action-index activate service-template service-template-name</i> • <i>action-index map attribute-to-service table parameter-map-name</i> <p>Example:</p> <p>The following example shows how a class-map with a service-template has to be applied:</p> <pre>Device(config-class-control-policymap)# 10 class cls_mac do-until-failure Device(config-action-control-policymap)# 10 activate service-template svcl</pre> <p>Example:</p> <p>The following example shows how a parameter map has to be applied (service template is already associated with the parameter map 'param' while creating it):</p> <pre>Device(config-action-control-policymap)#1 map attribute-to-service table param</pre> | |
| Step 5 | <p>end</p> <p>Example:</p> <pre>Device(config-action-control-policymap)# end</pre> | Exits configuration mode. |
| Step 6 | <p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 7 | <p>wireless profile policy <i>wlan-policy-profile-name</i></p> <p>Example:</p> <pre>Device(config)# wireless profile policy wlan-policy-profilename</pre> | <p>Configures a wireless policy profile.</p> <p>Caution Do not configure aaa-override for native profiling under a named wireless profile policy. Native profiling is applied at a lower priority than AAA policy. If aaa-override is enabled, the AAA policies will override native profile policy.</p> |
| Step 8 | <p>description <i>profile-policy-description</i></p> <p>Example:</p> <pre>Device(config-wireless-policy)# description "default policy profile"</pre> | Adds a description for the policy profile. |
| Step 9 | <p>dhcp-tlv-caching</p> <p>Example:</p> | Configures DHCP TLV caching on a WLAN. |

| | Command or Action | Purpose |
|----------------|--|---|
| | Device (config-wireless-policy) # dhcp-tlv-caching | |
| Step 10 | http-tlv-caching Example: Device (config-wireless-policy) # http-tlv-caching | Configures client HTTP TLV caching on a WLAN. |
| Step 11 | subscriber-policy-name <i>policy-name</i> Example: Device (config-wireless-policy) # subscriber-policy-name polmap5 | Configures the subscriber policy name. |
| Step 12 | vlan <i>vlan-id</i> Example: Device (config-wireless-policy) # vlan 1 | Configures a VLAN name or VLAN ID. |
| Step 13 | no shutdown Example: Device (config-wireless-policy) # no shutdown | Saves the configuration. |

Configuring Native Profiling in Local Mode

To configure native profiling in the local mode, you must follow the steps described in [#unique_1904](#). In the policy profile, you must enable central switching as described in the step given below in order to configure native profiling.

Procedure

| | Command or Action | Purpose |
|---------------|---|----------------------------|
| Step 1 | central switching Example: Device (config-wireless-policy) # central switching | Enables central switching. |

Verifying Native Profile Configuration

Use the following **show** commands to verify the native profile configuration:

```
Device# show wireless client device summary
```

```
Active classified device summary
MAC Address      Device-type      User-role
Protocol-map
-----
```

```

1491.82b8.f94b   Microsoft-Workstation   sales
          9
1491.82bc.2fd5   Windows7-Workstation   sales
          41

```

```
Device# show wireless client device cache
```

```
Cached classified device info
```

| MAC Address | Device-type | User-role |
|---------------------|-----------------------|-----------|
| 2477.031b.aal8 9 | Microsoft-Workstation | |
| 30a8.db3b.a753 9 | Un-Classified Device | |
| 4400.1011.e8b5 9 | Un-Classified Device | |
| 980c.a569.7dd0 | Un-Classified Device | |

```
Device# show wireless client mac-address 4c34.8845.e32c detail | s
```

```
Session Manager:
```

```
Interface :
```

```

IIF ID       : 0x90000002
Device Type  : Microsoft-Workstation
Protocol Map : 0x000009
Authorized   : TRUE
Session timeout : 1800
Common Session ID: 78380209000000174BF2B5B9
Acct Session ID : 0
Auth Method Status List

```

```
Method : MAB
```

```

SM State      : TERMINATE
Authen Status : Success

```

```
Local Policies:
```

```

Service Template : wlan_svc_C414.3CCA.0A51 (priority 254)
Absolute-Timer   : 1800

```

```
Server Policies:
```

```
Resultant Policies:
```

```

Filter-ID      : acl-auto
Input QOS      : in_qos
Output QOS     : out_qos
Idle timeout   : 60 sec
VLAN           : 10
Absolute-Timer : 1000

```

Use the following **show** command to verify the class map details for a class map name:

```

Device# show class-map type control subscriber name test
Class-map          Action                               Exec Hit Miss Comp
-----          -
match-any test     match day Monday                                     0    0    0    0
match-any test     match join-time-of-day 8:00 18:00                 0    0    0    0

```

```
Key:
```

```
"Exec" - The number of times this line was executed
```

```
"Hit" - The number of times this line evaluated to TRUE
```

```
"Miss" - The number of times this line evaluated to FALSE
```

```
"Comp" - The number of times this line completed the execution of its condition without a need to continue on to the end
```

