



Configuring Application Visibility and Control

- [Finding Feature Information, on page 1](#)
- [Information About Application Visibility and Control, on page 1](#)
- [Supported AVC Class Map and Policy Map Formats, on page 2](#)
- [Prerequisites for Application Visibility and Control, on page 4](#)
- [Guidelines for Inter-Device Roaming with Application Visibility and Control, on page 5](#)
- [Restrictions for Application Visibility and Control, on page 5](#)
- [How to Configure Application Visibility and Control, on page 6](#)
- [Monitoring Application Visibility and Control, on page 20](#)
- [Examples: Application Visibility and Control, on page 21](#)
- [Additional References for Application Visibility and Control, on page 24](#)
- [Feature History and Information For Application Visibility and Control, on page 25](#)

Finding Feature Information

Your software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Information About Application Visibility and Control

Application Visibility and Control (AVC) classifies applications using deep packet inspection techniques with the Network-Based Application Recognition engine, and provides application-level visibility and control (QoS) in wireless networks. After the applications are recognized, the AVC feature enables you to either drop, mark, or police the data traffic.

AVC is configured by defining a class map in a QoS client policy to match a protocol.

Using AVC, we can detect more than 1000 applications. AVC enables you to perform real-time analysis and create policies to reduce network congestion, costly network link usage, and infrastructure upgrades.



Note You can view list of 30 applications in Top Applications in Monitor Summary section of the UI.

Traffic flows are analyzed and recognized using the NBAR2 engine at the access point. For more information about the NBAR2 Protocol Library, see http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html. The specific flow is marked with the recognized protocol or application, such as WebEx. This per-flow information can be used for application visibility using Flexible NetFlow (FNF).

AVC QoS actions are applied with AVC filters in both upstream and downstream directions. The QoS actions supported for upstream flow are drop, mark, and police, and for downstream flow are mark and police. AVC QoS is applicable only when the application is classified correctly and matched with the class map filter in the policy map. For example, if the policy has a filter based on an application name, and the traffic has also been classified to the same application name, then the action specified for this match in the policy will be applied. For all QoS actions, refer [Supported AVC Class Map and Policy Map Formats, on page 2](#).

Application Visibility and Control Protocol Packs

Protocol packs are a means to distribute protocol updates outside the controller software release trains, and can be loaded on the controller without replacing the controller software.

The Application Visibility and Control Protocol Pack (AVC Protocol Pack) is a single compressed file that contains multiple Protocol Description Language (PDL) files and a manifest file. A set of required protocols can be loaded, which helps AVC to recognize additional protocols for classification on your network. The manifest file gives information about the protocol pack, such as the protocol pack name, version, and some information about the available PDLs in the protocol pack.

The AVC Protocol Packs are released to specific AVC engine versions. You can load a protocol pack if the engine version on the controller platform is the same or higher than the version required by the protocol pack.

Supported AVC Class Map and Policy Map Formats

Supported AVC Class Map Format

Class Map Format	Class Map Example	Direction
match protocol <i>protocol name</i>	<code>class-map match-any webex-class match protocol webex-media</code>	Both upstream and downstream
match protocol attribute category <i>category-name</i>	<code>class-map match-any IM match protocol attribute category instant-messaging</code>	Both upstream and downstream
match protocol attribute sub-category <i>sub-category-name</i>	<code>class-map match-any realtimeconferencing match protocol attribute sub-category voice-video-chat-collaboration</code>	Both upstream and downstream

Class Map Format	Class Map Example	Direction
match protocol attribute application-group <i>application-group-name</i>	<pre>class-map match-any skype match protocol attribute application-group skype-group</pre>	Both upstream and downstream
Combination filters	<pre>class-map match-any webex-class match protocol webex match dscp 45 match wlan user-priority 6</pre>	Upstream only

Supported AVC Policy Format

Policy Format	QoS Action
Upstream client policy based on match protocol filter	Mark, police, and drop
Downstream client policy based on match protocol filter	Mark and police

The following table describes the detailed AVC policy format with an example:

AVC Policy Format	AVC Policy Example	Direction
Basic set	<pre>policy-map webex-policy class webex-class set dscp ef //or set up,cos</pre>	Upstream and downstream
Basic police	<pre>policy-map webex-policy class webex-class police 5000000</pre>	Upstream and downstream
Basic set and police	<pre>policy-map webex-policy class webex-class set dscp ef //or set up,cos police 5000000</pre>	Upstream and downstream
Multiple set and police including default	<pre>policy-map webex-policy class webex-class set dscp af31 //or set up,cos police 4000000 class class-webex-category set dscp ef //or set up,cos police 6000000 class class-default set dscp <></pre>	Upstream and downstream

AVC Policy Format	AVC Policy Example	Direction
Hierarchical police	<pre> policy-map webex-policy class webex-class police 500000 service-policy client-in-police-only policy-map client-in-police-only class webex-class police 100000 class class-webex-category set dscp ef //or set up,cos police 600000 police 200000 </pre>	Upstream and downstream
Hierarchical set and police	<pre> policy-map webex-policy class class-default police 150000 service policy client-up-child policy-map webex-policy class webex-class police 100000 set dscp ef class class-webex-category police 200000 set dscp af31 </pre>	
Drop action	<p>Any of the above examples apply to this format with this additional example:</p> <pre> policy-map webex-policy class webex-class drop class netflix set dscp ef //or set up,cos police 6000000 class class-default set dscp <> </pre>	Upstream only

Prerequisites for Application Visibility and Control

- The access points should be AVC capable.
- For the control part of AVC (QoS) to work, the application visibility feature with FNF has to be configured.

Guidelines for Inter-Device Roaming with Application Visibility and Control

Follow these guidelines to prevent clients from getting excluded due to malformed QoS policies:

- When a new QoS policy is added to the device, a QoS policy with the same name should be added to other device within the same roam or mobility domain.
- When a device is loaded with a software image of a later release, the new policy formats are supported. If you have upgraded the software image from an earlier release to a later release, you should save the configuration separately. When an earlier release image is loaded, some QoS policies might show as not supported, and you should restore those QoS policies to supported policy formats.

Restrictions for Application Visibility and Control

- AVC is supported only on the following access points:
 - Cisco Aironet 1260 Series Access Points
 - Cisco Aironet 1600 Series Access Points
 - Cisco Aironet 2600 Series Access Point
 - Cisco Aironet 2600 Series Wireless Access Points
 - Cisco Aironet 2700 Series Access Point
 - Cisco Aironet 3500 Series Access Points
 - Cisco Aironet 3600 Series Access Points
- AVC is not supported on Cisco Aironet 702W, 702I (128 M memory), and 1530 Series Access Points.
- Dropping or marking of the data traffic (control part) is not supported for software Release 3.3.
- Dropping or marking of the data traffic (control part) is supported in software Release 3E.
- Only the applications that are recognized with application visibility can be used for applying QoS control.
- Multicast traffic classification is not supported.
- Only the applications that are recognized with App visibility can be used for applying QoS control.
- IPv6 including ICMPv6 traffic classifications are not supported.
- Datalink is not supported for NetFlow fields for AVC.
- The following commands are not supported for AVC flow records:
 - **collect flow username**
 - **collect interface { input | output }**
 - **collect wireless client ipv4 address**
 - **match interface { input | output }**
 - **match transport igmp type**
- The template timeout cannot be modified on exporters configured with AVC. Even if the template timeout value is configured to a different value, only the default value of 600 seconds is used.

- For the username information in the AVC-based record templates, ensure that you configure the options **records** to get the user MAC address to username mapping. For more information, refer [Creating a Flow Exporter \(Optional\)](#), on page 9.
- When there is a mix of AVC-enabled APs such as 3600, and non-AVC-enabled APs such as 1140, and the chosen policy for the client is AVC-enabled, the policy will not be sent to the APs that cannot support AVC.
- Only ingress AVC statistics are supported. The frequency of statistics updates depends on the number of clients loaded at the AP at that time. Statistics are not supported for very large policy format sizes.
- The total number of flows for which downstream AVC QoS supported per client is 1000.
- The maximum number of flows supported for Catalyst 3850 Series Switch is 48 K.
- These are some class map and policy map-related restrictions. For supported policy formats, see [Supported AVC Class Map and Policy Map Formats](#), on page 2.
 - AVC and non-AVC classes cannot be defined together in a policy in a downstream direction. For example, when you have a class map with match protocol, you cannot use any other type of match filter in the policy map in the downstream direction.
 - Drop action is not applicable for the downstream AVC QoS policy.
 - Match protocol is not supported in ingress or egress for SSID policy.
- Google shares resources among several of their services because of which for some of the traffic it is not possible to say it is unique to one application. Therefore we added google-services for traffic that cannot be distinguished. The behavior you experience is expected.

How to Configure Application Visibility and Control

Configuring Application Visibility and Control (CLI)

To configure Application Visibility, follow these general steps:

1. Create a flow record by specifying keys and non-key fields to the flow.
2. Create an optional flow exporter by specifying the flow record as an option.
3. Create a flow monitor based on the flow record and flow exporter.
4. Configure WLAN to apply flow monitor in IPv4 input or output direction.

To configure Application Control, follow these general steps:

1. Create an AVC QoS policy.
2. Attach AVC QoS policy to the client in one of three ways: configuring WLAN, using ACS or ISE, or adding local policies.

Creating a Flow Record

By default, **wireless avc basic** (flow record) is available. When you click **Apply** from the GUI, then the record is mapped to the flow monitor.

Default flow record cannot be edited or deleted. If you require a new flow record, you need to create one and map it to the flow monitor from CLI.

SUMMARY STEPS

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *string*
4. **match ipv4 protocol**
5. **match ipv4 source address**
6. **match ipv4 destination address**
7. **match transport source-port**
8. **match transport destination-port**
9. **match flow direction**
10. **match application name**
11. **match wireless ssid**
12. **collect counter bytes long**
13. **collect counter packets long**
14. **collect wireless ap mac address**
15. **collect wireless client mac address**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow record <i>flow_record_name</i> Example: Device(config)# flow record record1 Device (config-flow-record)#	Enters flow record configuration mode.
Step 3	description <i>string</i> Example: Device (config-flow-record) # description IPv4flow	(Optional) Describes the flow record as a maximum 63-character string.
Step 4	match ipv4 protocol Example: Device (config-flow-record) # match ipv4 protocol	Specifies a match to the IPv4 protocol.
Step 5	match ipv4 source address Example: Device (config-flow-record) # match ipv4 source address	Specifies a match to the IPv4 source address-based field.

	Command or Action	Purpose
Step 6	match ipv4 destination address Example: Device (config-flow-record)# match ipv4 destination address	Specifies a match to the IPv4 destination address-based field.
Step 7	match transport source-port Example: Device (config-flow-record)# match transport source-port	Specifies a match to the transport layer source-port field.
Step 8	match transport destination-port Example: Device (config-flow-record)# match transport destination-port	Specifies a match to the transport layer destination-port field.
Step 9	match flow direction Example: Device (config-flow-record)# match flow direction	Specifies a match to the direction the flow was monitored in.
Step 10	match application name Example: Device (config-flow-record)# match application name	Specifies a match to the application name. Note This action is mandatory for AVC support, as this allows the flow to be matched against the application.
Step 11	match wireless ssid Example: Device (config-flow-record)# match wireless ssid	Specifies a match to the SSID name identifying the wireless network.
Step 12	collect counter bytes long Example: Device (config-flow-record)# collect counter bytes long	Specifies to collect counter fields total bytes.
Step 13	collect counter packets long Example: Device (config-flow-record)# collect counter bytes long	Specifies to collect counter fields total packets.
Step 14	collect wireless ap mac address Example: Device (config-flow-record)# collect wireless ap mac address	Specifies to collect the BSSID with MAC addresses of the access points that the wireless client is associated with.
Step 15	collect wireless client mac address Example:	Specifies to collect MAC address of the client on the wireless network.

	Command or Action	Purpose
	Device (config-flow-record) # collect wireless client mac address	
Step 16	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Creating a Flow Exporter (Optional)

You can create a flow export to define the export parameters for a flow. This is an optional procedure for configuring flow parameters.

SUMMARY STEPS

1. **configure terminal**
2. **flow exporter** *flow_exporter_name*
3. **description** *string*
4. **destination** *{hostname | ip-address}*
5. **transport udp** *port-value*
6. **option application-table timeout** *seconds* (optional)
7. **option usermac-table timeout** *seconds* (optional)
8. **end**
9. **show flow exporter**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow exporter <i>flow_exporter_name</i> Example: Device(config)# flow exporter record1 Device (config-flow-exporter)#	Enters flow exporter configuration mode.
Step 3	description <i>string</i> Example: Device(config-flow-exporter) # description IPv4flow	Describes the flow record as a maximum 63-character string.
Step 4	destination <i>{hostname ip-address}</i> Example: Device (config-flow-exporter) # destination 10.99.1.4	Specifies the hostname or IPv4 address of the system to which the exporter sends data.

	Command or Action	Purpose
Step 5	transport udp <i>port-value</i> Example: Device (config-flow-exporter) # transport udp 2	Configures a port value for the UDP protocol.
Step 6	option application-table timeout <i>seconds</i> (optional) Example: Device (config-flow-exporter) # option application-table timeout 500	(Optional) Specifies application table timeout option. The valid range is from 1 to 86400 seconds.
Step 7	option usermac-table timeout <i>seconds</i> (optional) Example: Device (config-flow-exporter) # option usermac-table timeout 1000	(Optional) Specifies wireless usermac-to-username table option. The valid range is from 1 to 86400 seconds.
Step 8	end Example: Device (config) # end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.
Step 9	show flow exporter Example: Device # show flow exporter	Verifies your configuration.
Step 10	end Example: Device (config) # end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record and a flow exporter.

SUMMARY STEPS

1. **configure terminal**
2. **flow monitor** *monitor-name*
3. **description** *description*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **cache timeout** { **active** | **inactive** } (Optional)
7. **end**
8. **show flow monitor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	flow monitor <i>monitor-name</i> Example: Device (config)# flow monitor flow-monitor-1	Creates a flow monitor and enters flow monitor configuration mode.
Step 3	description <i>description</i> Example: Device (config-flow-monitor)# description flow-monitor-1	Creates a description for the flow monitor.
Step 4	record <i>record-name</i> Example: Device (config-flow-monitor)# record flow-record-1	Specifies the name of a recorder that was created previously.
Step 5	exporter <i>exporter-name</i> Example: Device (config-flow-monitor)# exporter flow-exporter-1	Specifies the name of an exporter that was created previously.
Step 6	cache timeout { active inactive } (Optional) Example: Device (config-flow-monitor)# cache timeout active 1800 Device (config-flow-monitor)# cache timeout inactive 200	Specifies to configure flow cache parameters. You can configure for a time period of 1 to 604800 seconds (optional). Note To achieve optimal result for the AVC flow monitor, we recommend you to configure the inactive cache timeout value to be greater than 90 seconds.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.
Step 8	show flow monitor Example: Device # show flow monitor	Verifies your configuration.

Creating AVC QoS Policy

To create AVC QoS policy, perform these general steps:

1. Create a class map with match protocol filters.
2. Create a policy map.

3. Apply a policy map to the client in one of the following ways:
 - a. Apply a policy map over WLAN either from the CLI or GUI.
 - b. Apply a policy map through the AAA server (ACS server or ISE) from the CLI.
For more information, refer to the *Cisco Identity Services Engine User Guide* and *Cisco Secure Access Control System User Guide*.
 - c. Apply local policies either from the CLI or GUI.

Creating a Class Map

You need to create a class map before configuring any match protocol filter. The QoS actions such as marking, policing, and dropping can be applied to the traffic. The AVC match protocol filters are applied only for the wireless clients. Refer [8.0 protocol pack](#) for the protocols supported.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** *class-map-name*
3. **match protocol** { *application-name* | **attribute category** *category-name* | **attribute sub-category** *sub-category-name* | **attribute application-group** *application-group-name* }
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	class-map <i>class-map-name</i> Example: Device(config)# class-map webex-class	Creates a class map.
Step 3	match protocol { <i>application-name</i> attribute category <i>category-name</i> attribute sub-category <i>sub-category-name</i> attribute application-group <i>application-group-name</i> } Example: Device(config)# class-map webex-class Device(config-cmap)# match protocol webex-media Device(config)# class-map class-webex-category Device(config-cmap)# match protocol attribute category webex-media Device# class-map class-webex-sub-category	Specifies match to the application name, category name, subcategory name, or application group.

	Command or Action	Purpose
	<pre>Device(config-cmap)# match protocol attribute sub-category webex-media Device# class-map class-webex-application-group Device(config-cmap)# match protocol attribute application-group webex-media</pre>	
Step 4	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Creating a Policy Map

SUMMARY STEPS

1. **configure terminal**
2. **policy-map** *policy-map-name*
3. **class** [*class-map-name* | **class-default**]
4. **police** *rate-bps burst-byte* [**exceed-action** {**drop** | **policed-dscp-transmit**}]
5. **set** {**dscp** *new-dscp* | **cos** *cos-value*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>policy-map <i>policy-map-name</i></p> <p>Example:</p> <pre>Device(config)# policy-map webex-policy Device(config-pmap)#</pre>	<p>Creates a policy map by entering the policy map name, and enters policy-map configuration mode.</p> <p>By default, no policy maps are defined.</p> <p>The default behavior of a policy map is to set the DSCP to 0 if the packet is an IP packet and to set the CoS to 0 if the packet is tagged. No policing is performed.</p> <p>Note To delete an existing policy map, use the no policy-map <i>policy-map-name</i> global configuration command.</p>
Step 3	<p>class [<i>class-map-name</i> class-default]</p> <p>Example:</p> <pre>Device(config-pmap)# class-map webex-class Device(config-pmap-c)#</pre>	<p>Defines a traffic classification, and enters policy-map class configuration mode.</p> <p>By default, no policy map and class maps are defined.</p> <p>If a traffic class has already been defined by using the class-map global configuration command, specify its name for <i>class-map-name</i> in this command.</p>

	Command or Action	Purpose
		<p>A class-default traffic class is predefined and can be added to any policy. It is always placed at the end of a policy map. With an implied match any is included in the class-default class, all packets that have not already matched the other traffic classes will match class-default.</p> <p>Note To delete an existing class map, use the no class class-map-name policy-map configuration command.</p>
Step 4	<p>police <i>rate-bps burst-byte</i> [exceed-action {drop policed-dscp-transmit}]</p> <p>Example:</p> <pre>Device(config-pmap-c)# police 100000 80000 drop</pre>	<p>Defines a policer for the classified traffic.</p> <p>By default, no policer is defined.</p> <ul style="list-style-type: none"> For <i>rate-bps</i>, specify an average traffic rate in bits per second (b/s). The range is 8000 to 10000000000. For <i>burst-byte</i>, specify the normal burst size in bytes. The range is 8000 to 1000000. (Optional) Specifies the action to take when the rates are exceeded. Use the exceed-action drop keywords to drop the packet. Use the exceed-action policed-dscp-transmit keywords to mark down the DSCP value (by using the policed-DSCP map) and to send the packet.
Step 5	<p>set {dscp <i>new-dscp</i> cos <i>cos-value</i>}</p> <p>Example:</p> <pre>Device(config-pmap-c)# set dscp 45</pre>	<p>Classifies IP traffic by setting a new value in the packet.</p> <ul style="list-style-type: none"> For dscp <i>new-dscp</i>, enter a new DSCP value to be assigned to the classified traffic. The range is 0 to 63.
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	<p>Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.</p>

What to do next

After creating your policy maps, attach the traffic policy or policies to an interface using the **service-policy** command.

Configuring Local Policies (CLI)

Configuring Local Policies (CLI)

To configure local policies, complete these procedures:

1. Create a service template.
2. Create an interface template.
3. Create a parameter map.

4. Create a policy map.
5. Apply a local policy on a WLAN.

Creating a Service Template (CLI)

SUMMARY STEPS

1. **configure terminal**
2. **service-template** *service-template-name*
3. **access-group** *acl_list*
4. **vlan** *vlan_id*
5. **absolute-timer** *seconds*
6. **service-policy qos** { **input** | **output** }
7. **end**

DETAILED STEPS

	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 cisco-phone-template Device(config-service-template)#	Enters service template configuration mode.
Step 3	access-group <i>acl_list</i> Example: Device(config-service-template)# access-group foo-acl	Specifies the access list to be applied.
Step 4	vlan <i>vlan_id</i> Example: Device(config-service-template)# vlan 100	Specifies VLAN ID. You can specify a value from 1 to 4094.
Step 5	absolute-timer <i>seconds</i> Example: Device(config-service-template)# absolute-timer 20	Specifies session timeout value for service template. You can specify a value from 1 to 65535.
Step 6	service-policy qos { input output } Example:	Configures QoS policies for the client.

	Command or Action	Purpose
	Device(config-service-template)# service-policy qos input foo-qos	
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Creating a Parameter Map (CLI)

Parameter map is preferred to use than class map.

SUMMARY STEPS

1. **configure terminal**
2. **parameter-map type subscriber attribute-to-service** *parameter-map-name*
3. *map-index* **map** { **device-type** | **mac-address** | **oui** | **user-role** | **username** } { **eq** | **not-eq** | **regex** *filter-name* }
4. **interface-template** *interface-template-name*
5. **end**

DETAILED STEPS

	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 <i>parameter-map-name</i> Example: Device(config)# parameter-map type subscriber attribute-to-service Aironet-Policy-para	Specifies the parameter map type and name.
Step 3	<i>map-index</i> map { device-type mac-address oui user-role username } { eq not-eq regex <i>filter-name</i> } Example: Device(config-parameter-map-filter)# 10 map device-type eq "WindowsXP-Workstation"	Specifies parameter map attribute filter criteria.
Step 4	interface-template <i>interface-template-name</i> Example: Device(config-parameter-map-filter-submode)#	Enters service template configuration mode.

	Command or Action	Purpose
	<code>interface-template cisco-phone-template</code> Device(config-parameter-map-filter-submode)#	
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Related Topics

[Information About Configuring Local Policies](#)

[Restrictions for Configuring Local Policies](#)

[Monitoring Local Policies](#)

[Examples: Local Policies Configuration](#)

Creating a Policy Map (CLI)

SUMMARY STEPS

1. **configure terminal**
2. **policy-map type control subscriber** *policy-map-name*
3. **event identity-update** { **match-all** | **match-first** }
4. *class_number* **class** { *class_map_name* | **always** } { **do-all** | **do-until-failure** | **do-until-success** }
5. *action-index* **map attribute-to-service table** *parameter-map-name*
6. **end**

DETAILED STEPS

	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 Aironet-Policy	Specifies the policy map type.
Step 3	event identity-update { match-all match-first } Example: Device(config-policy-map)# event identity-update match-all	Specifies match criteria to the policy map.
Step 4	<i>class_number</i> class { <i>class_map_name</i> always } { do-all do-until-failure do-until-success }	Configures the local profiling policy class map number and specifies how to perform the action. The class map configuration mode includes the following command options:

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config-class-control-policymap)# 1 class local_policy1_class do-until-success</pre>	<ul style="list-style-type: none"> • always—Executes without doing any matching but return success. • do-all—Executes all the actions. • do-until-failure—Execute all the actions until any match failure is encountered. This is the default value. • do-until-success—Execute all the actions until any match success happens.
Step 5	<p><i>action-index</i> map attribute-to-service table <i>parameter-map-name</i></p> <p>Example:</p> <pre>Device(config-policy-map)# 10 map attribute-to-service table Aironet-Policy-para</pre>	Specifies parameter map table to be used.
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Related Topics

- [Information About Configuring Local Policies](#)
- [Restrictions for Configuring Local Policies](#)
- [Monitoring Local Policies](#)
- [Examples: Local Policies Configuration](#)

Applying a Local Policy for a Device on a WLAN (CLI)

Before you begin

If the service policy contains any device type-based rules in the parameter map, ensure that the device classifier is already enabled.



Note You should use the **device classification** command to classify the device for it to be displayed correctly on the show command output.

SUMMARY STEPS

1. **configure terminal**
2. **wlan** *wlan-name*
3. **service-policy type control subscriber** *polycymapname*
4. **profiling local http** (optional)
5. **profiling radius http** (optional)
6. **no shutdown**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	wlan wlan-name Example: Device(config)# wlan wlan1	Enters WLAN configuration mode.
Step 3	service-policy type control subscriber policymapname Example: Device(config-wlan)# service-policy type control subscriber Aironet-Policy	Applies local policy to WLAN.
Step 4	profiling local http (optional) Example: Device(config-wlan)# profiling local http	Enables only profiling of devices based on HTTP protocol (optional).
Step 5	profiling radius http (optional) Example: Device(config-wlan)# profiling radius http	Enables profiling of devices on ISE (optional).
Step 6	no shutdown Example: Device(config-wlan)# no shutdown	Specifies not to shut down the WLAN.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Related Topics

[Information About Configuring Local Policies](#)

[Restrictions for Configuring Local Policies](#)

[Monitoring Local Policies](#)

[Examples: Local Policies Configuration](#)

Configuring WLAN to Apply Flow Monitor in IPV4 Input/Output Direction

SUMMARY STEPS

1. **configure terminal**
2. **wlan wlan-id**
3. **ip flow monitor monitor-name {input | output}**

4. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	wlan wlan-id Example: Device (config) # wlan 1	Enters WLAN configuration submode. For <i>wlan-id</i> , enter the WLAN ID. The range is 1 to 64.
Step 3	ip flow monitor monitor-name {input output} Example: Device (config-wlan) # ip flow monitor flow-monitor-1 input	Associates a flow monitor to the WLAN for input or output packets.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Monitoring Application Visibility and Control

Monitoring Application Visibility and Control (CLI)

This section describes the new commands for application visibility.

The following commands can be used to monitor application visibility on the and access points.

Table 1: Monitoring Application Visibility Commands on the

Command	Purpose
show avc client client-mac top n application [aggregate upstream downstream]	Displays information about top "N" applications for the given client MAC.
show avc wlan ssid top n application [aggregate upstream downstream]	Displays information about top "N" applications for the given SSID.
avc top user[enable disable]	Enables or disables the information about top "N" application.
show avc wlan wlan-id application app name topN [aggregate upstream downstream]	Displays to know network usage information on a per user basis within an application

show wlan id <i>wlan-id</i>	Displays information whether AVC is enabled or disabled on a particular WLAN.
show flow monitor <i>flow_monitor_name</i> cache	Displays information about flow monitors.
show wireless client mac-address <i>mac-address</i> service-policy { input output }	Displays information about policy mapped to the wireless clients.
show ip nbar protocol-discovery [interface <i>interface-type</i> <i>interface-number</i>] [stats { byte-count bit-rate packet-count max-bit-rate }] [protocol <i>protocol-name</i> top-nnumber]	Displays the statistics gathered by the NBAR Protocol Discovery feature. • (Optional) Enter keywords and arguments to fine-tune the statistics displayed. For more information on each of the keywords, refer to the show ip nbar protocol-discovery command in Cisco IOS Quality of Service Solutions Command Reference. Note When you configure NBAR, you must enable Protocol Discovery on the interface.
show policy-map target show policy-map show policy-map <i>policy-name</i> show policy-map interface <i>interface-type</i> <i>interface-number</i>	Displays information about policy map.

Table 2: Clearing Application Visibility Statistics Commands

Command	Purpose
clear avc client <i>mac</i> stats	Clears the statistics per client.
clear avc wlan <i>wlan-name</i> stats	Clears the statistics per WLAN.

Examples: Application Visibility and Control

Examples: Application Visibility Configuration

This example shows how to create a flow record, create a flow monitor, apply the flow record to the flow monitor, and apply the flow monitor on a WLAN:

```
Device# configure terminal
Device(config)# flow record fr_v4
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
```

```

Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow direction
Device(config-flow-record)# match application name
Device(config-flow-record)# match wireless ssid
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect wireless ap mac address
Device(config-flow-record)# collect wireless client mac address
Device(config)#end

Device# configure terminal
Device# flow monitor fm_v4
Device(config-flow-monitor)# record fr_v4
Device(config-flow-monitor)# cache timeout active 1800
Device(config)#end

Device(config)#wlan wlan1
Device(config-wlan)#ip flow monitor fm_v4 input
Device(config-wlan)#ip flow mon fm-v4 output
Device(config)#end

```

Examples: Application Visibility and Control QoS Configuration

This example shows how to create class maps with apply match protocol filters for application name, category, and subcategory:

```

Device# configure terminal
Device(config)# class-map cat-browsing
Device(config-cmap)# match protocol attribute category browsing
Device(config-cmap)#end

Device# configure terminal
Device(config)# class-map cat-fileshare
Device(config-cmap)# match protocol attribute category file-sharing
Device(config-cmap)#end

Device# configure terminal
Device(config)# class-map match-any subcat-terminal
Device(config-cmap)# match protocol attribute sub-category terminal
Device(config-cmap)#end

Device# configure terminal
Device(config)# class-map match-any webex-meeting
Device(config-cmap)# match protocol webex-meeting
Device(config-cmap)#end

```

This example shows how to create policy maps and define existing class maps for upstream QoS:

```

Device# configure terminal
Device(config)# policy-map test-avc-up
Device(config-pmap)# class cat-browsing
Device(config-pmap-c)# police 150000
Device(config-pmap-c)# set dscp 12
Device(config-pmap-c)#end

Device# configure terminal
Device(config)# policy-map test-avc-up

```

```
Device(config-pmap) # class cat-fileshare
Device(config-pmap-c) # police 1000000
Device(config-pmap-c) # set dscp 20
Device(config-pmap-c) #end

Device# configure terminal
Device(config) # policy-map test-avc-up
Device(config-pmap) # class subcat-terminal
Device(config-pmap-c) # police 120000
Device(config-pmap-c) # set dscp 15
Device(config-pmap-c) #end

Device# configure terminal
Device(config) # policy-map test-avc-up
Device(config-pmap) # class webex-meeting
Device(config-pmap-c) # police 50000000
Device(config-pmap-c) # set dscp 21
Device(config-pmap-c) #end
```

This example shows how to create policy maps and define existing class maps for downstream QoS:

```
Device# configure terminal
Device(config) # policy-map test-avc-down
Device(config-pmap) # class cat-browsing
Device(config-pmap-c) # police 200000
Device(config-pmap-c) # set dscp 10
Device(config-pmap-c) #end

Device# configure terminal
Device(config) # policy-map test-avc-up
Device(config-pmap) # class cat-fileshare
Device(config-pmap-c) # police 300000
Device(config-pmap-c) # set wlan user-priority 2
Device(config-pmap-c) # set dscp 20
Device(config-pmap-c) #end

Device# configure terminal
Device(config) # policy-map test-avc-up
Device(config-pmap) # class subcat-terminal
Device(config-pmap-c) # police 100000
Device(config-pmap-c) # set dscp 25
Device(config-pmap-c) #end

Device# configure terminal
Device(config) # policy-map test-avc-up
Device(config-pmap) # class webex-meeting
Device(config-pmap-c) # police 60000000
Device(config-pmap-c) # set dscp 41
Device(config-pmap-c) #end
```

This example shows how to apply defined QoS policy on a WLAN:

```
Device# configure terminal
Device(config) # wlan alpha
Device(config-wlan) # shut
Device(config-wlan) #end
Device(config-wlan) #service-policy client input test-avc-up
```

Example: Configuring QoS Attribute for Local Profiling Policy

```
Device(config-wlan)#service-policy client output test-avc-down
Device(config-wlan)#no shut
Device(config-wlan)#end
```

Example: Configuring QoS Attribute for Local Profiling Policy

The following example shows how to configure QoS attribute for a local profiling policy:

```
Device(config)# class-map type control subscriber match-all local_policy1_class
Device(config-filter-control-classmap)# match device-type android
Device(config)# service-template local_policy1_template
Device(config-service-template)# vlan 40
Device(config-service-template)# service-policy qos output local_policy1
Device(config)# policy-map type control subscriber local_policy1
Device(config-event-control-policymap)# event identity-update match-all
Device(config-class-control-policymap)# 1 class local_policy1_class do-until-success
Device(config-action-control-policymap)# 1 activate service-template local_policy1_template
Device(config)# wlan open_auth 9
Device(config-wlan)# client vlan VLAN40
Device(config-wlan)# service-policy type control subscriber local_policy1
```

Additional References for Application Visibility and Control

Related Documents

Related Topic	Document Title
System management commands	<i>System Management Command Reference Guide, Cisco IOS XE Release 3SE (Cisco WLC 5700 Series)</i>
Flexible NetFlow configuration	<i>Flexible NetFlow Configuration Guide, Cisco IOS XE Release 3SE (Cisco WLC 5700 Series)</i>
Flexible NetFlow commands	<i>Flexible NetFlow Command Reference, Cisco IOS XE Release 3SE (Cisco WLC 5700 Series)</i>
QoS configuration	<i>QoS Configuration Guide, Cisco IOS XE Release 3E (Cisco WLC 5700 Series)</i>
QoS commands	<i>QoS Command Reference, Cisco IOS XE Release 3E (Cisco WLC 5700 Series)</i>

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
All supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/support

Feature History and Information For Application Visibility and Control

Release	Feature Information
Cisco IOS XE 3.3SE	This feature was introduced.
Cisco IOS XE 3E	AVC control with QoS was introduced.

