# Packet Classification Commands

This chapter describes the commands used for QoS packet classification.

To use commands of this module, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using any command, contact your AAA administrator for assistance.

For detailed information about Packet Classification concepts, configuration tasks and examples, see the Configuring Modular QoS Service Packet Classification and Marking chapter in the *Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers*.


- class (policy-map), on page 3
- class-map, on page 5
- end-class-map, on page 7
- end-policy-map, on page 8
- exceed-action, on page 9
- hw-module qos-mode pwhe-aggregate-shaper , on page 12
- match cac, on page 14
- match cos, on page 15
- match vlan, on page 17
- match ethertype, on page 19
- match flow-key, on page 21
- match fr-de, on page 23
- match frame-relay dlci, on page 24
- match access-group, on page 26
- match destination-address, on page 28
- match discard-class, on page 30
- match dscp, on page 32
- match mpls experimental topmost, on page 35
- match precedence, on page 37
- match protocol, on page 40
- match qos-group, on page 42
- set fr-de, on page 44
- set discard-class, on page 46
- set dscp, on page 48
- set ip encapsulation, on page 50
- set mpls experimental, on page 52

- set precedence, on page 54
- ntp(ipv4/ipv6), on page 56
- dscp(ntp), on page 57
- dscp precedence(ntp), on page 60
- shape average, on page 62
- show qos pwhe-aggregrate-shaper, on page 64
- show qos-ea interface, on page 66
- show qos status interface, on page 73

# class (policy-map)

To specify the name of the class whose policy you want to create or change, use the **class** command in policy map configuration mode. To remove a class from the policy map, use the **no** form of this command.

**class** [**type qos**] {*class-name* | **class-default**}
**no class** [**type qos**] {*class-name* | **class-default**}

## Syntax Description

| | |
|---|---|
| **type qos** | (Optional) Specifies a quality-of-service (QoS) class. |
| *class-name* | Name of the class for which you want to configure or modify policy. |
| **class-default** | Configures the default class. |

## Command Default

No class is specified.

Type is QoS when not specified.

## Command Modes

Policy map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

Within a policy map, the **class (policy-map)** command can be used to specify the name of the class whose policy you want to create or change. The policy map must be identified first.

To identify the policy map (and enter the required policy map configuration mode), use the **policy-map** command before you use the **class (policy-map)** command. After you specify a policy map, you can configure the policy for new classes or modify the policy for any existing classes in that policy map.

## Usage Guidelines

The class name that you specify in the policy map ties the characteristics for that class—that is, its policy—to the class map and its match criteria, as configured using the **class-map** command.

The **class-default** keyword is used for configuring default classes. It is a reserved name and cannot be used with user-defined classes. It is always added to the policy map (type qos) even if the class is not configured. For example, the following configuration shows that the class has not been configured, but the running configuration shows ‘class class-default’.

```
RP/0/RSP0/CPU0:router(config)# policy-map pm1
RP/0/RSP0/CPU0:router(config-pmap)# end-policy-map
RP/0/RSP0/CPU0:router(config)# end
!
RP/0/RSP0/CPU0:router# show running-config
!
policy-map pm1
 class class-default
 !
 end-policy-map
!
```

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to create a policy map called policy1, which is defined to shape class1 traffic at 30 percent and default class traffic at 20 percent.

```
RP/0/RSP0/CPU0:router(config)# class-map class1
RP/0/RSP0/CPU0:router(config-cmap)# match precedence 3
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:routerconfig-pmap-c)# shape average 100 mbps
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# class class-default
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average 50 mbps
```

The default class is used for packets that do not satisfy configured match criteria for class1. Class1 must be defined before it can be used in policy1, but the default class can be directly used in a policy map, as the system defines it implicitly.

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# class-map

To define a traffic class and the associated rules that match packets to the class, use the **class-map** command in Global Configuration mode mode. To remove an existing class map from the router, use the **no** form of this command.

**class-map** [**type** **[traffic | qos]**] [**match-all**] [**match-any**] *class-map-name*
**no** **class-map** [**type** **[traffic | qos**]] [**match-all**] [**match-any**] *class-map-name*

## Syntax Description

| | |
|---|---|
| **type qos** | (Optional) Specifies a quality-of-service (QoS) class-map. |
| **traffic** | (Optional) Specifies traffic type class-map. |
| **match-all** | (Optional) Specifies a match on all of the match criteria. |
| **match-any** | (Optional) Specifies a match on any of the match criteria. This is the default. |
| *class-map-name* | Name of the class for the class map. The class name is used for the class map and to configure policy for the class in the policy map. The class name can be a maximum of 63 characters, must start with an alphanumeric character, and in addition to alphanumeric characters, can contain any of the following characters: . _ @ $ % + \| # : ; - = |

## Command Default

Type is QoS when not specified.

## Command Modes

Global Configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **class-map** command specifies the name of the class for which you want to create or modify class map match criteria. Use of this command enables class map configuration mode in which you can enter any **match** command to configure the match criteria for this class. Packets arriving on the interface are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

These commands can be used in a class map:

- **match access-group**
- **match atm**
- **match [not] cos**
- **match destination-address**
- **match [not] discard-class**
- **match [not] dscp**
- **match frame-relay dlci**
- **match [not] mpls experimental topmost**
- **match [not] precedence**
- **match precedence**
- **match [not] protocol**
- **match [not] qos-group**

- **match source-address**
- **match vlan** *vlan-id*
- **match vpls**

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to specify class101 as the name of a class and defines a class map for this class. The packets that match the access list 101 are matched to class class101.

```
RP/0/RSP0/CPU0:router(config)# class-map class101
RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 101
```

# end-class-map

To end the configuration of match criteria for the class and to exit class map configuration mode, use the **end-class-map** command in class map configuration mode.

**end-class-map**

## Syntax Description

This command has no keywords or arguments.

## Command Default

No default behavior or values

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to end the class map configuration and exit class map configuration mode:

```
RP/0/RSP0/CPU0:router(config)# class-map class101
RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 101
RP/0/RSP0/CPU0:router(config-cmap)# end-class-map
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |

# end-policy-map

To end the configuration of a policy map and to exit policy map configuration mode, use the **end-policy-map** command in policy map configuration mode.

**end-policy-map**

## Syntax Description

This command has no keywords or arguments.

## Command Default

No default behavior or values

## Command Modes

Policy map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to end the policy map configuration and exit policy map configuration mode.

```
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 250
RP/0/RSP0/CPU0:router(config-pmap-c)# set precedence 3
RP/0/RSP0/CPU0:router(config-pmap)# end-policy-map
```

## Related Commands

| Command | Description |
|---|---|
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# exceed-action

To configure the action to take on packets that exceed the rate limit, use the **exceed-action** command in policy map police configuration mode. To remove an exceed action from the policy-map, use the **no** form of this command.

**exceed-action** [**drop** | **set** *options* | **transmit**]
**no exceed-action** [**drop** | **set** *options* | **transmit**]

## Syntax Description

| | |
|---|---|
| **drop** | (Optional) Drops the packet. |
| **set** *options* | Configures the specified packet properties. Replace *options* with one of the following keywords or keyword arguments:<br>• **atm-clp value** —Sets the cell loss priority (CLP) bit.<br>• **cos [inner]** *value* —Sets the class of service value. Range is 0 to 7.<br>• **cos** *value* —Sets the class of service value. Range is 0 to 7.<br>• **dei** —Sets the drop eligible indicator (DEI). Can be 0 or 1.<br>• **discard-class** *value* —Sets the discard class value. Range is 0 to 7.<br>• **dscp** *value* —Sets the differentiated services code point (DSCP) value and sends the packet. See Table 1: IP DSCP Reserved Keywords, on page 33 for a list of valid values.<br>• **dscp** [**tunnel**] *value* —Sets the differentiated services code point (DSCP) value and sends the packet. See Table 1: IP DSCP Reserved Keywords, on page 33 for a list of valid values. With the **tunnel** keyword, the DSCP is set in the outer header.<br>• **fr-de** *value* —Sets the Frame Relay discard eligible (DE) bit on the Frame Relay frame then transmits that packet. In congested environments, frames with the DE bit set to 1 are discarded before frames with the DE bit set to 0. The frame relay DE bit has only one bit and has only two settings, 0 or 1. The default DE bit setting is 0.<br>• **mpls experimental** {**topmost** | **imposition**} *value* —Sets the experimental (EXP) value of the Multiprotocol Label Switching (MPLS) packet topmost label or imposed label. Range is 0 to 7.<br>• **precedence** *precedence* —Sets the IP precedence and sends the packet. See Table 2: IP Precedence Values and Names, on page 38 for a list of valid values.<br>• **precedence** [**tunnel**] *precedence* —Sets the IP precedence and sends the packet. See Table 2: IP Precedence Values and Names, on page 38 for a list of valid values. With the **tunnel** keyword, the precedence is set in the outer header.<br>• **qos-group** *value* —Sets the QoS group value.<br>• **qos-group** *value* —Sets the QoS group value. Range is 0 to 63.<br>• **srp-priority** *value* —Sets the Spatial Reuse Protocol (SRP) priority. Range is 0 to 7. |
| **transmit** | (Optional) Transmits the packets. |

## Command Default

By default, if no action is configured on a packet that exceeds the rate limit, the packet is dropped.

## Command Modes

Policy map police configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 4.0.0 | The **set dei** keyword was added. |

## Usage Guidelines

For more information regarding the traffic policing feature, see the police rate command.

The **set dei** action in policy maps is supported on 802.1ad packets for:

- Ingress and egress
- Layer 2 subinterfaces
- Layer 2 main interfaces
- Layer 3 main interfaces

**Note** The set DEI action is ignored for traffic on interfaces that are not configured for 802.1ad encapsulation.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

In this example for MPLS, traffic policing is configured to drop traffic that exceeds the rate limit:

```
RP/0/RSP0/CPU0:router(config)# class-map class1
RP/0/RSP0/CPU0:router(config-cmap)# match mpls experimental topmost 0
RP/0/RSP0/CPU0:router(config-cmap)# exit

RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 250 kbps burst 50
RP/0/RSP0/CPU0:router(config-pmap-c-police)# exceed-action drop
RP/0/RSP0/CPU0:router(config-pmap-c-police)# exit
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit

RP/0/RSP0/CPU0:router(config)# interface pos 0/5/0/0
RP/0/RSP0/CPU0:router(config-if) service-policy input policy1
```

In this example, the police rate is set to 5 Mbps. Conforming traffic is marked with a DEI value of 0; traffic that exceeds the police rate is marked with a DEI value of 1.

```
RP/0/RSP0/CPU0:router(config)# policy-map 1ad-mark-dei
RP/0/RSP0/CPU0:router(config-pmap)# class c1
RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 5 mbps
RP/0/RSP0/CPU0:router(config-pmap-c-police)# conform-action set dei 0
```

```
RP/0/RSP0/CPU0:router(config-pmap-c-police)# exceed-action set dei 1
RP/0/RSP0/CPU0:router(config-pmap-c-police)# end-policy-map
```

## Related Commands

| Command | Description |
|---|---|
| conform-action | Configures the action to take on packets that conform to the rate limit. |
| police rate | Configures traffic policing and enters policy map police configuration mode. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| show policy-map interface | Displays policy configuration information for all classes configured for all service policies on the specified interface. |
| violate-action | Configures the action to take on packets that violate the rate limit. |

# hw-module qos-mode pwhe-aggregate-shaper

To enable co-existence mode, use the **hw-module qos-mode pwhe-aggregate-shaper** command in Global Configuration mode. To disable co-existence mode, use the **no** form of this command.

**Note** If you are running either on Cisco IOS XR or Cisco IOS XR 64 bit, use the **hw-module qos-mode pwhe-aggregate-shaper** command in Global configuration mode.

**hw-module** {**all** | **location** *node-id*} **qos-mode pwhe-aggregate-shaper sub-interface** {**egress** | **ingress**} {**non-queuing** | **queuing**}
**no hw-module** {**all** | **location** *node-id*} **qos-mode pwhe-aggregate-shaper sub-interface** {**egress** | **ingress**}

## Syntax Description

| | |
|---|---|
| **all** | Specifies all location. |
| **location** *node-id* | Specifies location of a particular node. The *node-id* argument is entered in the rack/slot/module notation. |
| **qos-mode** | Configures QoS mode in hardware module ports. For more information on various QoS modes, use the question mark (?) online help function. |
| **pwhe-aggregate-shaper** | Configures PWHE aggregate shaper parameters. |
| **sub-interface** | Configures PWHE subinterface QoS policy co-existence mode. |
| **egress** | Configures PWHE co-existence mode in egress direction. |
| **ingress** | Configures PWHE co-existence mode in ingress direction. |
| **non-queuing** | Configures co-existence non-queuing mode. |
| **queuing** | Configures co-existence queuing mode. |

## Command Default

None.

## Command Modes

Global Configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.1.2 | Support for Cisco IOS XR 64 bit bit was added. |
| Release 5.1.1 | This command was introduced. |

## Usage Guidelines

This command indicates that the PWHE main interface policies can co-exist with subinterface policies.

The mode change takes effect immediately on a line card (LC) if there is no PWHE main or subinterface policy on the LC, otherwise LC reload is required.

**Note** To avoid LC reload we recommend that the mode change is performed when there is no policy applied on any PWHE main or subinterfaces.

The co-existence mode with subinterface queuing policies is known as co-existence queuing mode.

The co-existence mode with subinterface non-queuing policies is known as co-existence non-queuing mode.

Adding, deleting and in-place modification of the PWHE aggregate shaper are allowed regardless of presence of subinterface policies.

**hw-module { all | location** *node-id* } **qos-mode pwhe-aggregate-shaper sub-interface ingress queuing** will not be rejected if there are LCs in the chassis which do not support ingress queuing.

Use the **show qos pwhe-aggregate-shaper** command to view current configured mode and operating mode.

**Task ID**

| Task ID | Operation |
|---|---|
| root-lr | read, write |

### Example

This example shows how to configure co-existence queuing mode in egress direction for all the line cards.

```
RP/0/RSP0/CPU0:router(config)# hw-module all qos-mode pwhe-aggregate-shaper sub-interface
egress queuing
```

# match cac

To identify specified call admission control (CAC) fields as a match criteria in a class map, use the **match cac** command in class map configuration mode. To remove a specified CAC field from the matching criteria for a class map, use the **no** form of this command.

**match cac** {**admitted** | **unadmitted**}**[ local ]**
**no match cac** {**admitted** | **unadmitted**}

## Syntax Description

| | |
|---|---|
| **admitted** | Specifies the packets admitted by CAC action as the match criteria in a class map. |
| **unadmitted** | Specifies the packets unadmitted by CAC action as the match criteria in a class map. |
| **local** | Specified the CAC local. |

## Command Default

No match criteria are specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 5.1.1 | This command was introduced. |

## Usage Guidelines

The **match cac** command specifies a call admission control field or action that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match cac** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish.

## Task ID

| Task ID | Operation |
|---|---|
| qos | read, write |

This example shows how to configure admit class for CAC.

```
RP/0/RSP0/CPU0:router(config)# class-map match-all video
RP/0/RSP0/CPU0:router(config-cmap)# match cac admitted local
RP/0/RSP0/CPU0:router(config-cmap)# exit
```

# match cos

To identify specified class of service (CoS) values as a match criteria in a class map, use the **match cos** command in class map configuration mode. To remove a specified CoS class value from the matching criteria for a class map, use the **no** form of this command.

**match** [**not**] **cos** {*cos-value* [*cos-value1 . . . cos-value7*] | **inner** *inner-cos-value* [*inner cos-value1 . . . inner cos-value7*]}
**no match** [**not**] **cos** {*cos-value* [*cos-value1 . . . cos-value7*] | **inner** *inner-cos-value* [*inner cos-value1 . . . inner cos-value7*]}

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *cos-value* | Identifier that specifies the exact value from 0 to 7. Up to eight CoS identifiers can be specified to match packets. |
| **inner** | (Optional) Specifies the inner CoS value in, for example, a QinQ configuration. |
| *inner-cos-value* | Identifier that specifies the exact value from 0 to 7. Up to eight inner CoS identifiers can be specified to match packets. |

## Command Default

No match criteria are specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match cos** command specifies a class of service that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match cos** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match cos** command in a class map, the values of subsequent match statements are added to the first **match cos** command.

The **match cos** command is supported on egress Layer 2 interfaces, Layer 2 subinterfaces, and Layer 3 physical interfaces. Layer 3 physical interfaces are supported, because it is possible for a Layer 3 interface to have underlying Layer 2 subinterfaces.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to configure the service policy called policy1 and attach service policy policy1 to an interface. In this example, class map cos146 evaluates all packets entering HundredGigE interface 0/7/0/0 .100 for class of service values of 1, 4, or 6. If the incoming packet has been marked

with any of these CoS values, the packet is queued to the class queue with the bandwidth setting of 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map cos146
RP/0/RSP0/CPU0:router(config-cmap)# match cos 1 4 6
RP/0/RSP0/CPU0:router(config-cmap)# exit

RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class cos146
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit

RP/0/RSP0/CPU0:router(config)# interface HundredGigE 0/7/0/0
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| set cos | Sets the Layer 2 class of service (CoS) value of an outgoing packet. |

# match vlan

To identify selected VLAN IDs as the match criteria for a class map, use the **match vlan** command in class map configuration mode. To remove VLAN ID match criteria from a class map, use the **no** form of this command.

**match vlan** [**inner**] *vlanid* [*vlanid1 . . . vlanid7*]
**no match vlan** [**inner**] *vlanid* [*vlanid1 . . . vlanid7*]

## Syntax Description

| | |
|---|---|
| **inner** | (Optional) Specifies the inner VLAN for the match in, for example, a QinQ configuration. |
| *vlanid* | VLAN identifier that specifies the exact value from 1 to 4094 or a range of values from 1 to 4094. Up to eight values can be specified in a match statement. |

## Command Default

No match criteria are specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match vlan** command specifies a VLAN ID that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match vlan** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match vlan** command in a class map, up to eight values of the subsequent match statements are added to the first **match vlan** command, exceeding which, the statement is rejected.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to configure the service policy called policy1 and attach service policy policy1 to an interface. In this example, class map vlan1 evaluates all packets entering GigabitEthernet interface 0/1/0/0 for VLAN IDs of 1234, 1698, and all the VLAN IDs in the range 3000 to 4000. If the incoming packet has been marked with any of these VLAN IDs, the packet is queued to the class queue with the bandwidth setting of 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map vlan1
RP/0/RSP0/CPU0:router(config-cmap)# match vlan 1234 1698 3000-4000
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class vlan1
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
```

```
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |

# match ethertype

To identify selected Ethernet type fields as the match criteria for a class map, use the **match ethertype** command in class map configuration mode. To remove the Ethernet type fields match criteria from a class map, use the **no** form of this command.

**match ethertype** *ethertype_field*
**no match ethertype** *ethertype_field*

## Syntax Description

*ethertype_field* Ethernet type field that specifies the Ethernet service. Values can be:

- <1536-65535>—Ethertype value; only 2054 (the value representing ARP) is supported.
- **arp** —Match on address resolution protocol.
- ipv4 (not supported).
- ipv6 (not supported).

**Caution**
If you specify any value other than the supported values, the system allows you to commit the configuration. However, if you try to commit a configuration that applies the policy-map (containing the invalid ethertype match) to an interface, the system rejects the commit action.

## Command Default

No match criteria are specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 4.1.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| qos | read, write |

### Example

This example shows how to configure Ethernet type:

```
RP/0/RSP0/CPU0:router(config)# class-map match-any arp_1
RP/0/RSP0/CPU0:router(config-cmap)# match ethertype arp
RP/0/RSP0/CPU0:router(config-cmap)# exit
```

**Related Commands**

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |

# match flow-key

To identify specified flow keys as a match criteria in a class map, use the **match flow-key** command in class map configuration mode. To remove a specified flow key from the matching criteria for a class map, use the **no** form of this command.

**match flow-key** [**5-tuple** | **dst-ip** | **src-ip** ] [**flow-cache** {**idle-timeout***period* | **none**}]
**no match flow-key**

## Syntax Description

| | |
|---|---|
| **5-tuple** | Specifies the 5 tuples to identify a flow. The 5 tuples required to define a flow are: source IP address, destination IP address, source port number, destination port number, and protocol. |
| **dst-ip** | Specifies the destination IP address of the flow mask. |
| **src-ip** | Specifies the source IP address of the flow mask. |
| **flow-cache** | Configures the flow-cache parameters. |
| **idle-timeout** | Specifies the maximum time of inactivity for a flow. |
| *period* | Specifies the idle timeout period in seconds. |
| **none** | Specifies that the flow does not expire. |

## Command Default

- Default flow mask is 5-tuple.
- Default idle timeout is 30s.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 5.1.1 | This command was introduced. |

## Usage Guidelines

The **match flow-key** command specifies a flow mask that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

## Task ID

| Task ID | Operation |
|---|---|
| qos | read, write |

This example shows how to configure the source IP address of the flow mask as the only flow tuple and a flow idle timeout of 200 seconds.

```
RP/0/RSP0/CPU0:router(config)# class-map match-all video1
RP/0/RSP0/CPU0:router(config-cmap)# match precedence 5
```

```
RP/0/RSP0/CPU0:router(config-cmap)# match flow-key src-ip flow-cache idle-timeout 200
RP/0/RSP0/CPU0:router(config-cmap)# exit
```

# match fr-de

To match packets on the basis of the Frame Relay discard eligibility (DE) bit setting, use the **match fr-de** command in class-map configuration mode. To remove the match criterion, use the **no** form of this command.

**match fr-de** *fr-de-bit-value*
**no match fr-de** *fr-de-bit-value*

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *fr-de-bit-value* | Specifies the Frame Relay DE bit. Value can be 1. |

## Command Default

Packets are not matched on the basis of the Frame Relay DE bit setting.

## Command Modes

Class map configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 4.0.0 | This command was introduced. |

## Usage Guidelines

This match criterion can be used under a subinterface (L2 PVC) using the **service-policy** command, and it is supported in the ingress direction only. This match criterion can also be used in hierarchical policy maps.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

The following example shows how to create a class called match-fr-de and match packets on the basis of the Frame Relay DE bit setting. Packets match Frame Relay DE bit 1.

```
RP/0/RSP0/CPU0:router(config)# class-map match-fr-de
RP/0/RSP0/CPU0:router(config-cmap)# match fr-de 1
RP/0/RSP0/CPU0:routerrouter(config-cmap)# end
```

To match Frame Relay DE bit 0, use this configuration:

```
RP/0/RSP0/CPU0:router(config)# class-map match-not-fr-de
RP/0/RSP0/CPU0:router(config-cmap)# match not fr-de 1
RP/0/RSP0/CPU0:router(config-cmap)# end
```

## Related Commands

| Command | Description |
|---|---|
| set fr-de, on page 44 | Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface. |

# match frame-relay dlci

To specify a Frame Relay packet data-link connection identifier (DLCI) number or number range as a match criterion in a class map, use the **match frame-relay dlci** command in class map configuration mode. To remove a previously specified DLCI number as a match criterion, use the **no** form of this command.

**match frame-relay dlci** [*DlciStartDlci-EndDlci*]
**no match frame-relay dlci** [*DlciStartDlci-EndDlci*]

## Syntax Description

| | |
|---|---|
| *Dlci* | A DLCI number associated with the packet. Range is from 16 to 1007. |
| *StartDlci-EndDlci* | A DLCI number range from 16 to 1007. Numbers are separated by a hyphen. |

## Command Default

No DLCI number is specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 4.0.0 | This command was introduced. |

## Usage Guidelines

The match criterion for the **match frame-relay dlci** command can be used only on hierarchical policy maps.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

The following example shows how to create the fr-dlci class map, and specify the Frame Relay DLCI number range 100-200 as a match criterion. Packets with DLCIs matching this criterion are placed in fr-dlci. In this example, class map fr-dlci evaluates all packets entering Packet-over-SONET/SDH (POS) interface 0/1/0/0.1 for DLCIs in the range from 100 through 200. If the incoming packet has been marked with the DLCI in the range from 100 through 200, the packet is queued to the class queue with the bandwidth setting of 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map fr-dlci
RP/0/RSP0/CPU0:router(config-cmap)# match frame-relay dlci 100-200
RP/0/RSP0/CPU0:router(config-cmap)# end
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class fr-dlci
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit

RP/0/RSP0/CPU0:router(config)# interface pos 0/1/0/0.1
RP/0/RSP0/CPU0:router(config)# service-policy output policy1

RP/0/RSP0/CPU0:router(config)# interface pos 0/1/0/0.1 point-to-point pvc 16
```

```
RP/0/RSP0/CPU0:router(config-subif)# pvc 16
RP/0/RSP0/CPU0:router(config-fr-vc)# service-policy output policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| match dscp, on page 32 | Configures a DSCP value as a match criterion. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| show frame-relay pvc | Displays statistics about Frame Relay permanent virtual circuits (PVCs). |

# match access-group

To identify a specified access control list (ACL) number as the match criteria for a class map, use the **match access-group** command in class map configuration mode. To remove ACL match criteria from a class map, use the **no** form of this command.

**match** [**not**] **access-group** {**ipv4** | **ipv6** | **ethernet-service**} *access-group-name*
**no match** [**not**] **access-group** {**ipv4** | **ipv6** | **ethernet-service**} *access-group-name*

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| **ipv4** | Specifies the name of the IPv4 access group to be matched. |
| **ipv6** | Specifies the name of the IPv6 access group to be matched. |
| **ethernet-service** | Specifies the name of the ethernet-service access group to be matched. |
| *access-group-name* | ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. |

## Command Default

By default, if neither IPv6 nor IPv4 is specified as the match criteria for a class map, IPv4 addressing is used.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 4.2.1 | Support for IPv6 ACLs with source port matching was added in MQC policy maps. |
| Release 6.0.1 | The **ethernet-service** keyword was introduced. |

## Usage Guidelines

For class-based features (such as marking, Modified Deficit Round Robin [MDRR], and policing), you define traffic classes based on match criteria, including ACLs and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class.

The **match access-group** command specifies an ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

Access Control Entries with TCP fields such as, SYN, ACK and FIN in the corresponding ACL are not supported.

To use the **match access-group** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. You can specify up to eight IPv4 and IPv6 ACLs in a match statement.

QoS classification based on the packet length or TTL (time to live) field in the IPv4 and IPv6 headers is not supported.

When an ACL list is used within a class-map, the deny action of the ACL is ignored and the traffic is classified based on the specified ACL match parameters.

**Note**

- You can provide multiple values in a configuration; only the first value is considered for the match criteria. Subsequent values indicated in the match statement are ignored for classification.
- The capture statements in an ethernet-service ACL are ignored.
- An ethernet-service ACL should have only permit statements. If there are any deny statements, the policy is rejected.
- If you specify a value for the **Ether-Type** keyword using the **match access-group ethernet-service** command, the value is ignored.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to specify a class map called map1 and configures map1 to be used as the match criteria for this class:

```
RP/0/RSP0/CPU0:router(config)# class-map map1
RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 map1

RP/0/RSP0/CPU0:router(config-cmap)# match access-group ethernet-service map1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# match destination-address

To identify a specific destination MAC address explicitly as a match criterion in a class map, use the **match destination-address** command in class map configuration mode. To remove a specific destination MAC address from the matching criteria for a class map, use the **no** form of this command.

**match destination-address mac** *address*
**no match destination-address mac** *address*

## Syntax Description

| | |
|---|---|
| **mac** | Specifies a MAC address. |
| *address* | Specifies a destination MAC address. |

## Command Default

No default behavior or values

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match destination-address** command specifies a destination address that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match destination-address** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match destination-address** command in a class map, only the last command entered applies.

The **match destination-address** command is supported only on an output service policy.

Layer 2 match criteria on a Layer 3 target, or Layer 3 match criteria on a Layer 2 target, is not allowed. The **match destination-address** command is supported on egress Layer 2 interfaces, Layer 2 subinterfaces, and Layer 3 physical interfaces. Layer 3 physical interfaces are supported, because it is possible for a Layer 3 interface to have underlying Layer 2 subinterfaces.

The command is allowed on a policy map that is attached to an Ethernet interface. The command is invalid on a policy that is attached to a Packet-over-SONET/SDH (POS) interface or a routed VLAN subinterface.

The match 48-bit MAC address is specified in xxxx.xxxx.xxxx format on L2VPN PE interfaces.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to match a destination MAC address:

```
RP/0/RSP0/CPU0:router(config)#class-map match-any A
RP/0/RSP0/CPU0:router(config-cmap)# match destination-address mac 000.f0d0.2356
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |

# match discard-class

To identify specific discard class values as a match criteria for a class map, use the **match discard-class** command in class map configuration mode. To remove specified discard class values from the matching criteria for a class map, use the **no** form of this command.

**match** [**not**] **discard-class** *discard-class-value* [*discard-class-value1 . . . discard-class-value7*]
**no match** [**not**] **discard-class** *discard-class-value* [*discard-class-value1 . . . discard-class-value7*]

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *discard-class-value* | Discard class identifier. You can specify up to eight discard class identifiers to match packets. Class identifiers are separated by spaces. Range is 0 to 7. |

## Command Default

No default behavior or values

## Command Modes

Class map configuration

## Command History

| Command | Description |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match discard-class** command specifies a discard class that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match discard-class** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match discard-class** command in a class map, the new values are added to the existing match statement.

The **match discard-class** command sets the match criteria for examining discard classes marked on the packet. Up to eight discard class values can be matched in one match statement. For example, **match discard-class 0 1 2 3 4 5 6 7** returns matches for discard class values 0, 1, 2, 3, 4, 5, 6, and 7. Only one of the seven values is needed to yield a match (OR operation).

The discard class value is used as a matching criterion only. The value has no mathematical significance. For instance, the discard class value 2 is not greater than 1. The value simply indicates that a packet marked with the discard class of 2 should be treated differently than a packet marked with a discard class value of 1.

The **match discard-class** command is not supported on the Asynchronous Transfer Mode (ATM) interfaces.

**Note** The **match discard-class** command is applied only for egress policies.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows a service policy called policy1 attached to an interface. In this example, class map discard class5 is created to evaluate all packets leaving GigabitEthernet 0/1/0/9 interface for a discard-class value of 5. Packets marked with the discard class value of 5 are queued to a class queue with the bandwidth setting 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map discard-class5
RP/0/RSP0/CPU0:router(config-cmap)# match discard-class 5
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class discard-class5
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy output policy1
```

# match dscp

To identify specific IP differentiated services code point (DSCP) values as match criteria for a class map, use the **match dscp** command in class map configuration mode. To remove a DSCP value from a class map, use the **no** form of this command.

**match dscp** {[**ipv4** | | **ipv6**] *dscp-value* [*dscp-value1* . . . *dscp-value7*] | [*min-value* - *max-value*]}
**no match dscp** {[**ipv4** | | **ipv6**] *dscp-value* [*dscp-value1* . . . *dscp-value7*] | [*min-value* - *max-value*]}

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| **ipv4** | (Optional) Specifies the IPv4 DSCP value. |
| **ipv6** | (Optional) Specifies the IPv6 DSCP value. |
| *dscp-value* | IP DSCP value identifier that specifies the exact value or a range of values. Range is 0 - 63. Up to eight IP DSCP values can be specified to match packets. Reserved keywords can be specified instead of numeric values. Table 1: IP DSCP Reserved Keywords, on page 33 describes the reserved keywords. |
| *min-value* | Lower limit of DSCP range to match. Value range is 0 - 63. |
| *max-value* | Upper limit of DSCP range to match. Value range is 0 - 63. |

## Command Default

Matching on IP Version 4 (IPv4) and IPv6 packets is the default.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 5.2.0 | The *min-value* and *max-value* variables were added. |

## Usage Guidelines

The **match dscp** command specifies a DSCP value that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match dscp** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match dscp** command in a class map, only the last command entered applies.

The **match dscp** command examines the higher-order six bits in the type of service (ToS) byte of the IP header. Only one of the eight values is needed to yield a match (OR operation).

The command supports only eight IP DSCP values. If you try to configure more match statements after all the eight values are matched, the statements get rejected.

The IP DSCP value is used as a matching criterion only. The value has no mathematical significance. For instance, the IP DSCP value 2 is not greater than 1. The value simply indicates that a packet marked with the

IP DSCP value of 2 should be treated differently than a packet marked with an IP DSCP value of 1. The treatment of these marked packets is defined by the user through the setting of policies in policy map class configuration mode.

***Table 1: IP DSCP Reserved Keywords***

| DSCP Value | Reserved Keyword |
|---|---|
| 0 | default |
| 10 | AF11 |
| 12 | AF12 |
| 14 | AF13 |
| 18 | AF21 |
| 20 | AF22 |
| 22 | AF23 |
| 26 | AF31 |
| 28 | AF32 |
| 30 | AF33 |
| 34 | AF41 |
| 36 | AF42 |
| 38 | AF43 |
| 46 | EF |
| 8 | CS1 |
| 16 | CS2 |
| 24 | CS3 |
| 32 | CS4 |
| 40 | CS5 |
| 48 | CS6 |
| 56 | CS7 |
| ipv4 | ipv4 dscp |
| ipv6 | ipv6 dscp |

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to configure the service policy called policy1 and attach service policy policy1 to an interface. In this example, class map dscp14 evaluates all packets entering Packet-over-SONET/SDH (POS) interface 0/1/0/0for an IP DSCP value of 14. If the incoming packet has been marked with the IP DSCP value of 14, the packet is queued to the class queue with the bandwidth setting of 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map dscp14
RP/0/RSP0/CPU0:router(config-cmap)# match dscp ipv4 14
RP/0/RSP0/CPU0:router(config-cmap)# exit

RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class dscp14
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit

RP/0/RSP0/CPU0:router(config)# interface pos 0/1/0/0
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy1
```

# match mpls experimental topmost

To identify specific three-bit experimental (EXP) field values in the topmost Multiprotocol Label Switching (MPLS) label as match criteria for a class map, use the **match mpls experimental topmost** command in class map configuration mode. To remove experimental field values from the class map match criteria, use the **no** form of the command.

**match** [**not**] **mpls experimental topmost** *exp-value* [*exp-value1 . . . exp-value7*]
**no match** [**not**] **mpls experimental topmost** *exp-value* [*exp-value1 . . . exp-value7*]

## Syntax Description

| | |
|---|---|
| **not** | **not** |
| *exp-value* | Experimental value that specifies the exact value from 0 to 7. Up to eight experimental values can be specified to match MPLS headers. |

## Command Default

No default behavior or values

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match mpls experimental topmost** command is used by the class map to identify MPLS experimental values matching on a packet.

To use the **match mpls experimental topmost** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match mpls experimental topmost** command in a class map, the new values are added to the existing match statement.

This command examines the three experimental bits contained in the topmost label of an MPLS packet. Up to eight experimental values can be matched in one match statement. For example, **match mpls experimental topmost 2 4 5 7** returns matches for experimental values of 2, 4, 5, and 7. Only one of the four values is needed to yield a match (OR operation).

The experimental values are used as a matching criterion only. The value has no mathematical significance. For instance, the experimental value 2 is not greater than 1. The value indicates that a packet marked with the experimental value of 2 should be treated differently than a packet marked with the EXP value of 1. The treatment of these different packets is defined by the user through the setting of QoS policies in policy map class configuration mode.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to configure the service policy called policy1 and attach service policy policy1 to an interface. In this example, class map mplsmap1 evaluates all packets entering

GigabitEthernet interface 0/1/0/9 for an MPLS experimental value of 1. If the incoming packet has been marked with the MPLS experimental value of 1, the packet is queued to the class queue with the bandwidth setting of 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map mplsmap1
RP/0/RSP0/CPU0:router(config-cmap)# match mpls experimental topmost 1
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class mplsmap1
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| match dscp, on page 32 | Identifies specific IP differentiated services code point (DSCP) values as match criteria for a class map. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| set mpls experimental, on page 52 | Sets the experimental (EXP) value of the Multiprotocol Label Switching (MPLS) packet topmost or imposition labels. |
| show policy-map interface | Displays policy configuration information for all classes configured for all service policies on the specified interface. |

# match precedence

To identify IP precedence values as match criteria, use the **match precedence** command in class map configuration mode. To remove precedence values from a class map, use the **no** form of this command.

**match** [**not**] **precedence** [**ipv4** | **ipv6**] *precedence-value* [*precedence-value1* . . . *precedence-value7*]
**no match** [**not**] **precedence** [**ipv4** | **ipv6**] *precedence-value* [*precedence-value1* . . . *precedence-value7*]

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| **ipv4** | (Optional) Specifies the IPv4 precedence value. |
| **ipv6** | (Optional) Specifies the IPv6 precedence value. |
| *precedence-value* | An IP precedence value identifier that specifies the exact value. Range is from 0 to 7. Reserved keywords can be specified instead of numeric values. Table 2: IP Precedence Values and Names, on page 38 describes the reserved keywords.<br>Up to eight precedence values can be matched in one match statement. |

## Command Default

Matching on both IP Version 4 (IPv4) and IPv6 packets is the default.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match precedence** command specifies a precedence value that is used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match precedence** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. If you specify more than one **match precedence** command in a class map, only the last command entered applies.

The **match precedence** command examines the higher-order three bits in the type of service (ToS) byte of the IP header. Up to eight precedence values can be matched in one match statement. For example, **match precedence ipv4 0 1 2 3 4 5 6 7** returns matches for IP precedence values of 0, 1, 2, 3, 4, 5, 6, and 7. Only one of the eight values is needed to yield a match (OR operation).

The precedence values are used as a matching criterion only. The value has no mathematical significance. For instance, the precedence value 2 is not greater than 1. The value simply indicates that a packet marked with the precedence value of 2 is different than a packet marked with the precedence value of 1. The treatment of these different packets is defined by the user through the setting of QoS policies in policy map class configuration mode.

This table lists the IP precedence value number and associated name in descending order of importance.

*Table 2: IP Precedence Values and Names*

| Value | Name |
|---|---|
| 0 | routine |
| 1 | priority |
| 2 | immediate |
| 3 | flash |
| 4 | flash-override |
| 5 | critical |
| 6 | internet |
| 7 | network |
| ipv4 | ipv4 precedence |
| ipv6 | ipv6 precedence |

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to configure the service policy called policy1 and attach service policy policy1 to an interface. In this example, class map ipprec5 evaluates all packets entering GigabitEthernet interface 0/1/0/9 for a precedence value of 5. If the incoming packet has been marked with the precedence value of 5, the packet is queued to the class queue with the bandwidth setting 300 kbps.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# class-map ipprec5
RP/0/RSP0/CPU0:router(config-cmap)# match precedence ipv4 5
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class ipprec5
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy  input policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| set precedence, on page 54 | Sets the precedence value in the IP header. |
| show policy-map interface | Displays policy configuration information for all classes configured for all service policies on the specified interface. |

# match protocol

To identify a specific protocol as the match criterion for a class map, use the **match protocol** command in class map configuration mode. To remove protocol-based match criteria from a class map, use the **no** form of this command.

**match** [**not**] **protocol** {*protocol-value* [*protocol-value1* **. . .** *protocol-value7*] | [*min-value* – *max-value*]}
**no match** [**not**] **protocol** {*protocol-value* [*protocol-value1* **. . .** *protocol-value7*] | [ *min-value* – *max-value*]}

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *protocol-value* | A protocol identifier. A single value for *protocol-value* (any combination of numbers and names) can be matched in one match statement. |
| *min-value* | Lower limit of protocol range to match. Value range is 0 - 255. |
| *max-value* | Upper limit of protocol range to match. Value range is 0 - 255. |

## Command Default

No default behavior or values

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 5.2.0 | The *min-value* and *max-value* variables were added. |

## Usage Guidelines

Definitions of traffic classes are based on match criteria, including protocols, access control lists (ACLs), input interfaces, QoS labels, and experimental (EXP) field values. Packets satisfying the match criteria for a class constitute the traffic for that class.

The **match protocol** command specifies the name of a protocol to be used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map. Available protocol names are listed in the table that follows.

The *protocol-value* argument supports a range of protocol numbers. After you identify the class, you may use the **match protocol** command to configure its match criteria.

***Table 3: Protocol Names and Descriptions***

| Name | Description |
|---|---|
| ahp | Authentication Header Protocol |
| eigrp | Cisco Enhanced Interior Gateway Routing Protocol |
| esp | Encapsulation Security Payload |

| Name | Description |
|---|---|
| gre | Cisco Generic Routing Encapsulation Tunneling |
| icmp | Internet Control Message Protocol |
| igmp | Internet Gateway Message Protocol |
| igrp | Cisco IGRP Routing protocol |
| ipinip | IP in IP tunneling |
| ipv4 | Any IPv4 protocol |
| ipv6 | Any IPv6 protocol |
| mpls | Any MPLS packet |
| nos | KA9Q NOS Compatible IP over IP Tunneling |
| ospf | Open Shortest Path First, Routing Protocol |
| pcp | Payload Compression Protocol |
| pim | Protocol Independent Multicast |
| sctp | Stream Control Transmission Protocol |
| tcp | Transport Control Protocol |
| udp | User Datagram Protocol |

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

In this example, all TCP packets belong to class class1:

```
RP/0/RSP0/CPU0:router(config)# class-map class1
RP/0/RSP0/CPU0:router(config-cmap)# match protocol tcp
```

# match qos-group

To identify specific quality-of-service (QoS) group values as match criteria in a class map, use the **match qos-group** command in class map configuration mode. To remove a specific QoS group value from the matching criteria for a class map, use the **no** form of this command.

**match** [**not**] **qos-group** [*qos-group-value 1 . . . qos-group-value8*]
**no match** [**not**] **qos-group**

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *qos-group-value* | QoS group value identifier that specifies the exact value from 0 to 63 or a range of values from 0 to 63. Up to eight values can be entered in one match statement. |

## Command Default

No match criteria are specified.

## Command Modes

Class map configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **match qos-group** command sets the match criteria for examining QoS groups marked on the packet. Up to eight QoS group values can be matched in one match statement. For example, **match qos-group 4 9 11 15 16 21 30 31** returns matches for QoS group values of 4, 9, 11, 15, 16, 21, 30, and 31. Only one of the eight values is needed to yield a match (OR operation).

The QoS group value is used as a matching criterion only. The value has no mathematical significance. For instance, the QoS group value 2 is not greater than 1. The value simply indicates that a packet marked with the QoS group value of 2 should be treated differently than a packet marked with a QoS group value of 1. The treatment of these different packets is defined using the **service-policy** command in policy map class configuration mode.

The QoS group setting is limited in scope to the local router. Typically, the QoS group is set on the local router and is used in conjunction with WRED or MDRR to give differing levels of service based on the group identifier.

The **match qos-group** command is supported only on egress policies.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows a service policy called policy1 attached to an interface. In this example, class map qosgroup5 will evaluate all packets leaving GigabitEthernet interface 0/1/0/9 for a QoS group value of 5. If the packet has been marked with the QoS group value of 5, the packet is queued to the class queue with the bandwidth setting 300 kbps.

```
RP/0/RSP0/CPU0:router(config)# class-map qosgroup5
RP/0/RSP0/CPU0:router(config-cmap)# match qos-group 5
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class qosgroup5
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth 300
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| set discard-class, on page 46 | Sets the discard class and Quality of Service (QoS) group identifiers on IP Version 4 (IPv4) or Multiprotocol Label Switching (MPLS) packets. |

# set fr-de

To change the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface, use the **set fr-de** command in policy map configuration mode. To remove the DE bit setting, use the **no** form of this command.

**set fr-de** [*fr-de-bit-value*]
**no set fr-de** [*fr-de-bit-value*]

## Syntax Description

| | |
|---|---|
| **not** | (Optional) Negates the specified match result. |
| *fr-de-bit-value* | (Optional) Specifies the Frame Relay DE bit. Value can be 0 or 1. Value can be 1. |

## Command Default

The default value is 0.

## Command Modes

Policy map configuration

## Command History

| Release | Modification |
|---|---|
| Release 4.0.0 | This command was introduced. |

## Usage Guidelines

To disable this command in a traffic policy, use the **no set fr-de** command in policy map configuration mode of the traffic policy.

If the DE bit is already set to 1, no changes are made to the frame.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

The following example shows how to set the DE bit using the **set fr-de** command in the traffic policy. The router sets the DE bit of outbound packets belonging to the ip-precedence class.

```
RP/0/RSP0/CPU0:router(config)# class-map ip-precedence
RP/0/RSP0/CPU0:router(config-cmap)# match precedence 0 1
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map set-de
RP/0/RSP0/CPU0:router(config-pmap)# class ip-precedence
RP/0/RSP0/CPU0:router(config-pmap-c)# set fr-de 1
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1
RP/0/RSP0/CPU0:router(config-if)# no ip address
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RSP0/CPU0:router(config-if)# interface serial 0/1/0/0.1 point-to-point
RP/0/RSP0/CPU0:router(config-subif)# ip address 10.1.1.1 255.255.255.252
```

```
RP/0/RSP0/CPU0:router(config-subif)# pvc 16
RP/0/RSP0/CPU0:router(config-fr-vc)# service-policy output set-de
```

## Related Commands

| Command | Description |
|---|---|
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# set discard-class

To set the discard class and Quality of Service (QoS) group identifiers on IP Version 4 (IPv4) or Multiprotocol Label Switching (MPLS) packets, use the **set discard-class** command in policy map class configuration mode. To leave the discard-class values unchanged, use the **no** form of this command.

**set discard-class** *discard-class-value*
**no set discard-class** *discard-class-value*

## Syntax Description

| | |
|---|---|
| *discard-class-value* | Discard class ID. An integer from 0 to 7, to be marked on the packet. |

## Command Default

No default behavior or values

## Command Modes

Policy map class configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

The **set discard-class** command associates a discard class ID with a packet. After the discard class is set, other QoS services such as Modified Deficit Round Robin (MDRR) and Weighted Random Early Detection (WRED) can operate on the bit settings.

Discard-class indicates the discard portion of the per hop behavior (PHB). The **set discard-class** command is typically used in Pipe mode. Discard-class is required when the input PHB marking is used to classify packets on the output interface.

The discard-class values can be used to specify the type of traffic that is dropped when there is congestion.

**Note** Marking of the discard class has only local significance on a node.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to set the discard class value to 5 for packets that match the MPLS experimental bits 1:

```
RP/0/RSP0/CPU0:router(config)# class-map cust1
RP/0/RSP0/CPU0:router(config-cmap)# match mpls experimental topmost 1
RP/0/RSP0/CPU0:router(config-cmap)# exit
RP/0/RSP0/CPU0:router(config)# policy-map policy2
RP/0/RSP0/CPU0:router(config-pmap)# class cust1
RP/0/RSP0/CPU0:router(config-pmap-c)# set discard-class 5
```

```
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy2
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| service-policy (interface) | Attaches a policy map to an input interface or output interface to be used as the service policy for that interface. |

# set dscp

To mark a packet by setting the IP differentiated services code point (DSCP) in the type of service (ToS) byte, use the **set dscp** command in policy-map class configuration mode. To remove a previously set DSCP value, use the **no** form of this command.

**set dscp** [**tunnel**] *dscp-value*
**no set dscp** [**tunnel**] *dscp-value*

## Syntax Description

| | |
|---|---|
| **tunnel** | (Optional) Sets the DSCP on the outer IP header. This command is available on Layer 3 interfaces in the ingress direction. |
| | (Optional) Sets the DSCP on the outer IP header for IPsec tunnels. |
| *dscp-value* | Number from 0 to 63 that sets the DSCP value. Reserved keywords can be specified instead of numeric values. Table 1: IP DSCP Reserved Keywords, on page 33 describes the reserved keywords. |

## Command Default

No default behavior or values

## Command Modes

Policy map class configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 3.9.1 | The **tunnel** keyword on Layer 3 interfaces in the ingress direction was added. |

## Usage Guidelines

After the DSCP bit is set, other quality-of-service (QoS) services can then operate on the bit settings.

The network gives priority (or some type of expedited handling) to marked traffic. Typically, you set the DSCP value at the edge of the network (or administrative domain); data then is queued based on the DSCP value. Modified Deficit Round Robin (MDRR) can speed up handling for high DSCP traffic at congestion points. Weighted Random Early Detection (WRED) ensures that high DSCP traffic has lower loss rates than other traffic during times of congestion.

Reserved keywords can be specified instead of numeric values. Table 1: IP DSCP Reserved Keywords, on page 33 describes the reserved keywords.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

In this example, the DSCP ToS byte is set to 8 in the policy map called policy1. All packets that satisfy the match criteria of class1 are marked with the DSCP value of 8. The network configuration determines how packets are marked.

```
RP/0/RSP0/CPU0:router (config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# set dscp 8
```

## Related Commands

| Command | Description |
|---|---|
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| service-policy (interface) | Attaches a policy map to an input interface or output interface to be used as the service policy for that interface. |
| set precedence, on page 54 | Sets the precedence value in the IP header. |
| show policy-map interface | Displays policy configuration information for all classes configured for all service policies on the specified interface. |

# set ip encapsulation

To set traffic class imposition for SRv6 encapsulation, use the **set ip encapsulation** command in policy map class configuration mode.

**set ip encapsulation class-of-service** *cos-value*

## Syntax Description

| | |
|---|---|
| **class-of-service** | Specifies the class of service. |
| *cos-value* | Specifies the number that sets the SRv6 encapsulation in the IP header. The value ranges from 0 to 63. |

## Command Default

None

## Command Modes

Policy map class configuration

## Command History

| Release | Modification |
|---|---|
| Release 24.2.1 | This command was introduced. |

## Usage Guidelines

- The **set ip encapsulation class-of-service** command is not allowed in egress QoS policies.
- The **set qos-group** action cannot be used in conjunction with the **set ip encapsulation class-of-service** command.
- The IP encapsulation marking (**set ip encapsulation class-of-service**) and the MPLS experimental imposition marking ( **set mpls experimental**) features are mutually exclusive. Both the configurations are not allowed at the same time.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to set the traffic class imposition for SRv6 encapsulations.

```
RP/0/RSP0/CPU0:router(config)# class-map match-any class1
RP/0/RSP0/CPU0:router(config-pmap)# match protocol ipv4
RP/0/RSP0/CPU0:router(config-pmap-c)# match access-group ipv4 10
RP/0/RSP0/CPU0:router(config-cmap-c)# class class-default
RP/0/RSP0/CPU0:router(config-cmap-c)# end-class-map

RP/0/RSP0/CPU0:router(config)# policy-map SRv6
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# set ip encapsulation class-of-service 55
RP/0/RSP0/CPU0:router(config-cmap-c)# class class-default
RP/0/RSP0/CPU0:router(config-cmap-c)# end-policy-map
```

```
RP/0/RSP0/CPU0:router(config)# interface HundredGigE 0/0/0/24
RP/0/RSP0/CPU0:router(config-if)# service-policy input SRv6
```

# set mpls experimental

To set the experimental (EXP) value of the Multiprotocol Label Switching (MPLS) packet topmost or imposition labels, use the **set mpls experimental** command in policy map configuration mode. To leave the EXP value unchanged, use the **no** form of this command.

**set mpls experimental** {**imposition** | **topmost**} *exp-value*
**no set mpls experimental** {**imposition** | **topmost**} *exp-value*

## Syntax Description

| | |
|---|---|
| **imposition** | Specifies to set the EXP value of the imposition label. |
| **topmost** | Specifies to set the EXP value of the topmost label. |
| *exp-value* | Value of the MPLS packet label. Range is 0 to 7. |

## Command Default

No MPLS experimental value is set

## Command Modes

Policy map class configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

After the MPLS experimental bits are set, other QoS services such as Modified Deficit Round Robin (MDRR) and Weighted Random Early Detection (WRED) then operate on the bit settings.

The network gives priority (or some type of expedited handling) to the marked traffic through the application of MDRR or WRED at points downstream in the network. Typically, the MPLS experimental value is set at the edge of the network (or administrative domain) and queuing is acted on it thereafter. MDRR can speed up handling for high-priority traffic at congestion points. WRED ensures that high-priority traffic has lower loss rates than other traffic during times of congestion.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to set the MPLS experimental to 5 for packets that match access list 101:

```
RP/0/RSP0/CPU0:router(config)# class-map class1
RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 acl101
RP/0/RSP0/CPU0:router(config-cmap)# exit

RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# set mpls experimental topmost 5
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit
```

```
RP/0/RSP0/CPU0:router(config)# interface HundredGigE 0/1/0/0
RP/0/RSP0/CPU0:router(config-if)# service-policy output policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| service-policy (interface) | Attaches a policy map to an input interface or output interface to be used as the service policy for that interface. |

# set precedence

To set the precedence value in the IP header, use the **set precedence** command in policy map class configuration mode. To leave the precedence value unchanged, use the **no** form of this command.

**set precedence** [**tunnel**] *value*
**no set precedence** [**tunnel**] *value*

## Syntax Description

| | |
|---|---|
| **tunnel** | (Optional) Sets the IP precedence on the outer IP header. This command is available on Layer 3 interfaces in the ingress direction. |
| *value* | Number or name that sets the precedence bits in the IP header. Range is from 0 to 7. Reserved keywords can be specified instead of numeric values. Table 2: IP Precedence Values and Names, on page 38 describes the reserved keywords. |

## Command Default

No default behavior or values

## Command Modes

Policy map class configuration

## Command History

| | |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 3.9.1 | The **tunnel** keyword on Layer 3 interfaces in the ingress direction was added. |

## Usage Guidelines

Precedence can be set using a number or corresponding name. After IP Precedence bits are set, other QoS services such as Modified Deficit Round Robin (MDRR) and Weighted Random Early Detection (WRED) then operate on the bit settings.

The network gives priority (or some type of expedited handling) to the marked traffic through the application of MDRR or WRED at points downstream in the network. IP precedence can be set at the edge of the network (or administrative domain) and have queueing act on it thereafter. MDRR can speed handling for high-precedence traffic at congestion points. WRED ensures that high-precedence traffic has lower loss rates than other traffic during times of congestion.

The mapping from keywords such as 0 (routine) and 1 (priority) to a precedence value is useful only in some instances. That is, the use of the precedence bit is evolving. You can define the meaning of a precedence value by enabling other features that use the value. In the case of high-end Internet QoS, IP precedences can be used to establish classes of service that do not necessarily correspond numerically to better or worse handling in the network.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example shows how to set the IP precedence to 5 (critical) for packets that match the access control list named customer1:

```
RP/0/RSP0/CPU0:router(config)# class-map class1
RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 customer1
RP/0/RSP0/CPU0:router(config-cmap)# exit

RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# set precedence 5
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
RP/0/RSP0/CPU0:router(config-pmap)# exit

RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/CPU0:router(config-if)# service-policy output policy1
```

## Related Commands

| Command | Description |
|---|---|
| class-map, on page 5 | Defines a traffic class and the associated rules that match packets to the class. |
| policy-map | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| service-policy (interface) | Attaches a policy map to an input interface or output interface to be used as the service policy for that interface. |

# ntp(ipv4/ipv6)

To configure DSCP for source address (IPv4/IPv6) to mark NTP packets. The NTP packets are treated according to DSCP marking.

{**ntp ipv4** | **ntp ipv6**}

## Syntax Description

| | |
|---|---|
| **DSCP** | Differentiated Services Code Point (DSCP) is a field in an IP packet that enables service(different levels) to be allocated to network traffic. DSCP is the combination of IP Precedence and Type of Service fields. |
| **Set Precendence** | setting levels for IP packets. |

## Command Modes

NTP Configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 5.3.2 | This command was introduced |

## Usage Guidelines

It sets NTP packets with DSCP field and NTP packets can be V4 and V6 based respectively it set DSCP/TOS field in v4 or v6 IP headers.

## Task ID

| Task ID | Operation |
|---|---|
| IP Services | read, write |

### Example

```
RP/0/RSP0/CPU0:router(config)#ntp ipv4 ?
  dscp        Set IP DSCP (DiffServ CodePoint)
  precedence  Set precedence

RP/0/RSP0/CPU0:router(config)#ntp ipv6 ?
  dscp        Set IP DSCP (DiffServ CodePoint)
  precedence  Set precedence
```

# dscp(ntp)

To configure DSCP for source address (IPv4/IPv6) to mark NTP packets and these NTP packets are treated according to DSCP marking.

{**ntp ipv4** | **ipv6** } *dscp*

## Syntax Description

| | |
|---|---|
| **<0-63>** | Differentiated services codepoint value |
| **af11** | Match packets with AF11 dscp (001010) |
| **af12** | Match packets with AF12 dscp (001100) |
| **af13** | Match packets with AF13 dscp (001110) |
| **af21** | Match packets with AF21 dscp (010010) |
| **af22** | Match packets with AF22 dscp (010100) |
| **af23** | Match packets with AF23 dscp (010110) |
| **af31** | Match packets with AF31 dscp (011010 |
| **af32** | Match packets with AF32 dscp (011100) |
| **af33** | Match packets with AF33 dscp (011110) |
| **af41** | Match packets with AF41 dscp (100010) |
| **af42** | Match packets with AF42 dscp (100100) |
| **af43** | Match packets with AF43 dscp (100110) |
| **cs1** | Match packets with CS1(precedence 1) dscp (001000) |
| **cs2** | Match packets with CS2(precedence 2) dscp (010000) |
| **cs3** | Match packets with CS3(precedence 3) dscp (011000) |
| **cs4** | Match packets with CS4(precedence 4) dscp (100000) |
| **cs5** | Match packets with CS5(precedence 5) dscp (101000) |
| **cs6** | Match packets with CS6(precedence 6) dscp (110000) |
| **cs7** | Match packets with CS7(precedence 7) dscp (111000) |
| **default** | Match packets with default dscp (000000) |
| **ef** | Match packets with EF dscp (101110) |

## Command Default

The default IPv4/IPv6 packets are matched with default dscp (000000).

**Command Modes**

NTP Configuration mode.

**Command History**

| Release | Modification |
|---|---|
| Release 5.3.2 | This command was introduced. |

**Usage Guidelines**

Use **DSCP** level to configure levels for the NTP packets at IP layer.

**Task ID**

| Task ID | Operation |
|---|---|
| IP Services | read, write |

**Example**

```
RP/0/RSP0/CPU0:router(config-ntp)#ntp ipv4 dscp
 <0-63>   Differentiated services codepoint value
  af11     Match packets with AF11 dscp (001010)
  af12     Match packets with AF12 dscp (001100)
  af13     Match packets with AF13 dscp (001110)
  af21     Match packets with AF21 dscp (010010)
  af22     Match packets with AF22 dscp (010100)
  af23     Match packets with AF23 dscp (010110)
  af31     Match packets with AF31 dscp (011010)
  af32     Match packets with AF32 dscp (011100)
  af33     Match packets with AF33 dscp (011110)
  af41     Match packets with AF41 dscp (100010)
  af42     Match packets with AF42 dscp (100100)
  af43     Match packets with AF43 dscp (100110)
  cs1      Match packets with CS1(precedence 1) dscp (001000)
  cs2      Match packets with CS2(precedence 2) dscp (010000)
  cs3      Match packets with CS3(precedence 3) dscp (011000)
  cs4      Match packets with CS4(precedence 4) dscp (100000)
  cs5      Match packets with CS5(precedence 5) dscp (101000)
  cs6      Match packets with CS6(precedence 6) dscp (110000)
  cs7      Match packets with CS7(precedence 7) dscp (111000)
  default  Match packets with default dscp (000000)
  ef       Match packets with EF dscp (101110)

RP/0/RSP0/CPU0:router(config-ntp)#ntp ipv6 dscp
 <0-63>   Differentiated services codepoint value
  af11     Match packets with AF11 dscp (001010)
  af12     Match packets with AF12 dscp (001100)
  af13     Match packets with AF13 dscp (001110)
  af21     Match packets with AF21 dscp (010010)
  af22     Match packets with AF22 dscp (010100)
  af23     Match packets with AF23 dscp (010110)
  af31     Match packets with AF31 dscp (011010)
  af32     Match packets with AF32 dscp (011100)
  af33     Match packets with AF33 dscp (011110)
  af41     Match packets with AF41 dscp (100010)
  af42     Match packets with AF42 dscp (100100)
  af43     Match packets with AF43 dscp (100110)
  cs1      Match packets with CS1(precedence 1) dscp (001000)
  cs2      Match packets with CS2(precedence 2) dscp (010000)
  cs3      Match packets with CS3(precedence 3) dscp (011000)
```

```
cs4      Match packets with CS4(precedence 4) dscp (100000)
cs5      Match packets with CS5(precedence 5) dscp (101000)
cs6      Match packets with CS6(precedence 6) dscp (110000)
cs7      Match packets with CS7(precedence 7) dscp (111000)
default  Match packets with default dscp (000000)
ef       Match packets with EF dscp (101110)
```

# dscp precedence(ntp)

To configure DSCP Prcedence for the IPv4/IPv6 NTP packets, enter Network Time Protocol (NTP) configuration mode and run NTP configuration commands, use the DSCP Prcedence(NTP) command.

{**ntp ipv4** | **ipv6**} *dscp*

## Syntax Description

| | |
|---|---|
| **<0-7>** | Precedence value |
| **critical** | Match packets with critical precedence (5) |
| **flash** | Match packets with flash precedence (3) |
| **flash-override** | Match packets with flash override precedence (4) |
| **immediate** | Match packets with immediate precedence (2) |
| **internet** | Match packets with internetwork control precedence (6) |
| **network** | Match packets with network control precedence (7) |
| **priority** | Match packets with priority precedence (1) |
| **routine** | Match packets with routine precedence (0) |

## Command Modes

NTP Configuration mode.

## Command History

| Release | Modification |
|---|---|
| Release 5.3.2 | This command was introduced. |

## Usage Guidelines

Use **Prcedence** values to set the configuration levels for the NTP packets at IP layer.

## Task ID

| Task ID | Operation |
|---|---|
| IP Services | read, write |

### Example

```
RP/0/RSP0/CPU0:router(config-ntp)#ntp ipv4 precedence ?
<0-7>           Precedence value
critical        Match packets with critical precedence (5)
flash           Match packets with flash precedence (3)
flash-override  Match packets with flash override precedence (4)
immediate       Match packets with immediate precedence (2)
internet        Match packets with internetwork control precedence (6)
network         Match packets with network control precedence (7)
priority        Match packets with priority precedence (1)
```

```
routine         Match packets with routine precedence (0)



RP/0/RSP0/CPU0:router(config-ntp)#ntp ipv6 precedence ?
<0-7>           Precedence value
critical        Match packets with critical precedence (5)
flash           Match packets with flash precedence (3)
flash-override  Match packets with flash override precedence (4)
immediate       Match packets with immediate precedence (2)
internet        Match packets with internetwork control precedence (6)
network         Match packets with network control precedence (7)
priority        Match packets with priority precedence (1)
routine         Match packets with routine precedence (0)
```

# shape average

To shape traffic to the indicated bit rate according to the algorithm specified, use the **shape average** command in policy map class configuration mode. To remove traffic shaping, use the **no** form of this command.

**shape average** {**percent** {*percentage* | **$***shape-rate* = *percentage*} | *rate* [*units*] | **per-thousand** *value* | **per-million** *value* | **$***shape-rate* = *rate* [*units*]}

## Syntax Description

| | |
|---|---|
| **percent** *percentage* | Specifies the interface bandwidth in percentage. Values can be from 1 to 100. |
| *rate* | Average shaping rate in the specified units. Values can be from 1 to 4294967295. |
| *units* | (Optional) Units for the bandwidth. Values can be:<br>• **Excess burst size**—values can be from 1 to 4294967295<br>• **bps**—bits per second (default)<br>• **gbps**—gigabits per second<br>• **kbps**—kilobits per second<br>• **mbps**—megabits per second |
| **per-thousand** *value* | Specifies shape rate as parts per thousand of the available bandwidth. |
| **per-million** *value* | Specifies shape rate as parts per million of the available bandwidth. |
| **$** | Specifies that a QoS shaper variable is configured. |
| *shape-rate* | QoS shaper variable name that can be parameterized. |

## Command Default

*units*: **bps**

## Command Modes

Policy map class configuration

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 5.2.0 | $*shape-rate* =*rate* option was added to support QoS shaper parameterization in BNG. |

## Usage Guidelines

For **shape average** commands in the child policy, the reference used for percentage parameters is relative to the maximum rate of the parent. If shaping or policing is not configured on the parent, then the parent inherits the interface rate.

If you have both shape and bandwidth configured for a class, ensure that the shape percent value is always greater than the percent value for bandwidth.

The shaper parameters, used to override the locally-configured values in BNG, can either be sent to BNG by the RADIUS server during connection establishment, as CISCO VSAs in an Access Accept message, or can be sent to BNG as part of the CoA messages.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

## Examples

This example sets traffic shaping to 50 percent of the parent shaper rate:

```
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average percent 50
```

This example shows how to set traffic shaping to 100000 kbps:

```
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average 100000 kbps
```

This example shows how to set the default value of QoS shape-rate variable (shaper4) as 100000 kbps, in BNG:

```
RP/0/RSP0/CPU0:router(config)# policy-map policy1
RP/0/RSP0/CPU0:router(config-pmap)# class class1
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average $shaper4 = 100000 kbps
```

# show qos pwhe-aggregate-shaper

To display current configured coexistence mode and coexistence operating modes, use the **show qos pwhe-aggregate-shaper** command in EXEC mode.

**show qos pwhe-aggregate-shaper** {**all** | **location** *node-id*}

## Syntax Description

| | |
|---|---|
| **all** | Specifies all locations. |
| **location** *node-id* | Specifies location of a particular node. The *node-id* argument is entered in the rack/slot/module notation. |

## Command Default

None.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 5.1.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| qos | read |

### Example

This example shows PW-HE coexistence mode and coexistence operating modes for location 0/1/CPU0.

```
RP/0/RSP0/CPU0:router# show qos  pwhe-aggregate-shaper location 0/1/CPU0

                     Configured                    Operational
Node           ingress          egress           ingress          egress
------------------------------------------------------------------------------
0/1/CPU0       disabled         disabled         disabled         disabled
```

This example shows PW-HE coexistence mode and coexistence operating modes for all locations.

```
RP/0/RSP0/CPU0:router# show qos pwhe-aggregate-shaper all

                     Configured                    Operational
Node           ingress          egress           ingress          egress
------------------------------------------------------------------------------
0/1/CPU0       non-queuing      queuing          non-queuing      queuing

 'all' line cards configured for ingress coexistence non-queuing
```

This example shows PW-HE coexistence mode and coexistence operating modes for all locations. In this example, line card (LC) at location 0/2/CPU0 is not in the chassis.

```
RP/0/RSP0/CPU0:router# show qos pwhe-aggregate-shaper all

                     Configured                          Operational
Node             ingress        egress               ingress          egress
-------------------------------------------------------------------------------
0/2/CPU0         queuing        non-queuing             ?               ?
0/1/CPU0         non-queuing    queuing               non-queuing      queuing

'?' means line card is offline - has not booted up or not inserted into the chassis
```

This example shows PW-HE coexistence mode and coexistence operating modes for all locations. In this example, LC at location 0/2/CPU0 does not support PW-HE QoS and LC at location 0/1/CPU0 supports PW-HE QoS, and both the LCs are active.

```
RP/0/RSP0/CPU0:router# show qos pwhe-aggregate-shaper all

                     Configured                      Operational
Node           ingress          egress           ingress          egress
-------------------------------------------------------------------------------
0/2/CPU0       queuing          non-queuing      N/A              N/A
0/1/CPU0       non-queuing      queuing          non-queuing      queuing

'N/A' means line card does not support PWHE QoS "
```

# show qos-ea interface

To display internal programming information for an interface, use the **show qos-ea interface** command in EXEC mode.

**show qos-ea interface** *type interface-path-id* { **input** | **output** } {**detail** | **member**} *interface-type interface-path-id* [**location***interface-path-id* ]

## Syntax Description

| | |
|---|---|
| *type* | Interface type.<br>For interface type **PW-Ether** or **PW-IW** provide interface handle number. The range is from 1 to 32768. |
| *interface-path-id* | Physical interface or virtual interface. Use the **show interfaces** command to see a list of configured interfaces. |
| **input** | Refers to policy applied in ingress direction. |
| **output** | Refers to policy applied egress direction. |
| **detail** | Displays detailed output. |
| **member** | Specifies member of bundle interface or pin-down (generic list) interface of PWHE interface. |
| **location** | (Optional) Specifies the location of the node. |

## Command Default

None.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 5.1 | This command was introduced. |
| Release 5.1.1 | The show outputs were updated for CAC and UBRL details.<br>PWHE Ethernet interface type **PW-Ether** and **PW-IW** were added. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| qos | read |

## Examples

This is the sample output which shows the QoS information on a Ten Gigabit Ethernet interface, applicable for the releases 5.3.2 onwards:

```
RP/0/RSP0/CPU0:tardis1-pwhe-spe#show qos-ea interface tenGigE 0/0/0/0/0 output
Interface: TenGigE0_0_0_0_0 output policy: EGRESS_CHILD_POLICY
```

```
Total number of classes: 8
Total number of UBRL classes: 0
Total number of CAC classes: 0
------------------------------------------------------
Policy name: EGRESS_CHILD_POLICY
Hierarchical depth 1
Interface type TenGigE
Interface rate 10000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x00000100
ul_ifh 0x00000000, ul_id 0x00000080
uidb index 0x0002
qos_ifh 0x8108020800002
Local port 0, NP 0
Policy map id 0x1420, format 17, uidb index 0x0002
------------------------------------------------------
Index 0 Level 0 Class name cp1 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 1
Queue limit 125000000 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E0B
Queue: Q-ID 0x000803c0 Stat ID(Commit/Excess/Drop): 0x000813C0/0x00000000/0x00A69700
------------------------------------------------------
Index 1 Level 0 Class name cp2 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 2
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E0C
Queue: Q-ID 0x000803c1 Stat ID(Commit/Excess/Drop): 0x000813C1/0x00000000/0x00A69704
------------------------------------------------------
Index 2 Level 0 Class name cp3 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 3
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E0D
Queue: Q-ID 0x000803c3 Stat ID(Commit/Excess/Drop): 0x000813C3/0x00000000/0x00A6970C
------------------------------------------------------
Index 3 Level 0 Class name cp4 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 4
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E0E
Queue: Q-ID 0x000803c4 Stat ID(Commit/Excess/Drop): 0x000813C4/0x00000000/0x00A69710
------------------------------------------------------
Index 4 Level 0 Class name cp5 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 5
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E0F
Queue: Q-ID 0x000803c5 Stat ID(Commit/Excess/Drop): 0x000813C5/0x00000000/0x00A69714
------------------------------------------------------
Index 5 Level 0 Class name cp6 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 6
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E10
Queue: Q-ID 0x000803c6 Stat ID(Commit/Excess/Drop): 0x000813C6/0x00000000/0x00A69718
------------------------------------------------------
```

```
Index 6 Level 0 Class name cp7 service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF
Stats flags: Queuing enabled
Node Config: Priority: 7
Queue limit 13937408 Guarantee 1
Node Result: Class-based stats:Stat ID 0x00C68E11
Queue: Q-ID 0x000803c7 Stat ID(Commit/Excess/Drop): 0x000813C7/0x00000000/0x00A6971C
------------------------------------------------------
Index 7 Level 0 Class name class-default service_id 0x0 Policy name EGRESS_CHILD_POLICY
Node flags: LEAF Q_LEAF DEFAULT DEFAULT-ALL
Stats flags: Queuing enabled
Node Config:
WFQ: BW/Sum of BW/Excess ratio: 0kbps/0kbps/900
Queue limit 13937408 Guarantee 0
Node Result: Class-based stats:Stat ID 0x00C68E12
Queue: Q-ID 0x000803c2 Stat ID(Commit/Excess/Drop): 0x000813C2/0x00000000/0x00A69708
```

This is the sample output of the show qos-ea interface command (Applicable until Release 5.1.1):

```
RP/0/RSP0/CPU0:router# show qos-ea interface gigabitEthernet 0/0/0/30 output
Interface: TenGigE0_3_0_34.3 output policy: two
Total number of classes: 18
------------------------------------------------------
Policy name: two
Hierarchical depth 2
Interface type VLAN Subif
Interface rate 10000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x0A001DC0
ul_ifh 0x00000000, ul_id 0x00000040
uidb index 0x002D
qos_ifh 0x15800040002d
Local port 34, NP 5
Policy map id 0x2008, format 15, uidb index 0x002D
------------------------------------------------------
Index 0 Level 0 Class name p1 service_id 0x0 Policy name two
Node flags: None
Stats flags: Queuing enabled
Node Config:
Shape: CIR/CBS/PIR/PBS: 0kbps/62500000B/5000000kbps/62500000B
WFQ: BW/Sum of BW/Excess ratio: 0kbps/0kbps/1
Node Result: Class-based stats:Stat ID 0x005102DD mode: 16Q
Queue: N/A Stat ID(Commit/Excess/Drop): 0x006E01E0/0x00000000/0x006E01E1
Index 1 Level 1 Class name c1 service_id 0x0 Policy name p16
------------------------------------------------------
Index 17 Level 0 Class name class-default service_id 0x0 Policy name two
Node flags: LEAF Q_LEAF DEFAULT DEFAULT-ALL
Stats flags: Queuing enabled
Node Config:
WFQ: BW/Sum of BW/Excess ratio: 0kbps/0kbps/1
Queue limit 62500000 Guarantee 0
Node Result: Class-based stats:Stat ID 0x005102F4
Queue: Q-ID 0x00030082 Stat ID(Commit/Excess/Drop): 0x006E028A/0x00000000/0x006E028B
```

This is the sample output of the show qos-ea interface command (Applicable from Release 5.1.1):

```
RP/0/# show qos-ea interface gigabitEthernet 0/1/0/0 input
Interface: GigabitEthernet0_1_0_0 input policy: premium_services (Flow Aware Policy)
Total number of classes: 5
Total number of UBRL classes: 0
Total number of CAC classes: 1
------------------------------------------------------
Policy name: premium_services
Hierarchical depth 2
```

```
Interface type GigE
Interface rate 1000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x060000C0
ul_ifh 0x00000000, ul_id 0x00000080
uidb index 0x0002
qos_ifh 0x8100000800002
Local port 0, NP 0
Policy map id 0x2018, format 16, uidb index 0x0002
------------------------------------------------------
Index 0 Level 0 Class name dscp_cs5 service_id 0x0 Policy name premium_services
Node flags: LEAF
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 100000kbps/1250000B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102DD
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001803/0x005102DD/0x005102DE/0x005102DF
------------------------------------------------------
Index 1 Level 0 Class name dscp_cs6 service_id 0x0 Policy name premium_services
Flow QoS Info: CAC Cac Action
CAC Flow Mask Mode: 5-Tuple (0x20)
CAC Flow Rate: 128 kbps
CAC Rate: 896 kbps
CAC Max Flow Count: 7
CAC Flow Age: 20 Seconds
Node flags: None
Stats flags:
Node Config: None
Node Result: Class-based stats:Stat ID 0x005102E0
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
------------------------------------------------------
Index 2 Level 1 Class name video_admitted service_id 0x0 Policy name video_flows
Flow QoS Info: CAC Admit Class
CAC Flow Mask Mode: 5-Tuple (0x20)
CAC Max Flow Count: 7
CAC Flow Age: 20 Seconds
CAC Reject Action: Redirect
CAC Flow Counter ID: 0xf41e61
CAC Flow Counter Value: 0
Number of CAC flows admitted: 0
Parent policy premium_services Class dscp_cs6
Node flags: LEAF
Stats flags:
Node Config: Mark
Node Result: Class-based stats:Stat ID 0x005102E1
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
------------------------------------------------------
Index 3 Level 1 Class name class-default service_id 0x0 Policy name video_flows
Flow QoS Info: CAC Redirect Class
Parent policy premium_services Class dscp_cs6
Node flags: LEAF DEFAULT
Stats flags:
Node Config: Mark
Node Result: Class-based stats:Stat ID 0x005102E2
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
------------------------------------------------------
Index 4 Level 0 Class name class-default service_id 0x0 Policy name premium_services
Node flags: LEAF DEFAULT DEFAULT-ALL
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 30000kbps/375000B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102E3
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
```

```
Police ID (Token/Conform/Exceed/Violate): 0x00001804/0x005102E3/0x005102E4/0x005102E5
------------------------------------------------------
```

This is the sample output of the show qos-ea interface command with the flow related information for CAC:

```
RP/0/# show qos-ea interface gigabitEthernet 0/1/0/0 input
Interface: GigabitEthernet0_1_0_0 input policy: premium-services (Flow Aware Policy)
Total number of classes: 3
Total number of UBRL classes: 0
Total number of CAC classes: 1
Interface rate 1000000 kbps
------------------------------------------------------
Policy name: premium-services
Hierarchical depth 1
Interface type GigE
Port Shaper rate 0 kbps
Interface handle 0x060000C0
ul_ifh 0x00000000, ul_id 0x00000080
uidb index 0x0002
qos_ifh 0x8100000800002
Local port 0, NP 0
Policy map id 0x2014, format 18, uidb index 0x0002
------------------------------------------------------
Index 0 Level 0 Class name prec5 service_id 0x0 Policy name premium-services
Node flags: LEAF
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 100000kbps/1250000B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102DD
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001803/0x005102DD/0x005102DE/0x005102DF
------------------------------------------------------
Index 1 Level 0 Class name video service_id 0x0 Policy name premium-services
Flow QoS Info: CAC Admit Class
CAC Flow Mask Mode: 5-Tuple (0x20)
CAC Flow Rate: 128 kbps
CAC Rate: 896 kbps
CAC Max Flow Count: 7
CAC Flow Age: 20 Seconds
CAC Reject Action: DROP
CAC Flow Counter ID: 0xf41e61
CAC Flow Counter Value: 0
Number of CAC flows admitted: 0
Node flags: LEAF
Stats flags: CAC Reject Drop enabled
Node Config: Mark
Node Result: Class-based stats:Stat ID 0x005102E0
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
------------------------------------------------------
Index 2 Level 0 Class name class-default service_id 0x0 Policy name premium-services
Node flags: LEAF DEFAULT DEFAULT-ALL
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 30000kbps/375000B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102E2
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001804/0x005102E2/0x005102E3/0x005102E4
------------------------------------------------------
```

This is the sample output of the show qos-ea interface command with the flow related information for UBRL:

```
RP/0/ # show qos-ea interface gigabitEthernet 0/1/0/0 input
Interface: GigabitEthernet0_1_0_0 input policy: voice_flow (Flow Aware Policy)
```

```
Total number of classes: 3
Total number of UBRL classes: 1
Total number of CAC classes: 0
----------------------------------------------------
Policy name: voice_flow
Hierarchical depth 1
Interface type GigE
Interface rate 1000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x060000C0
ul_ifh 0x00000000, ul_id 0x00000080
uidb index 0x0002
qos_ifh 0x8100000800002
Local port 0, NP 0
Policy map id 0x2008, format 16, uidb index 0x0002
----------------------------------------------------
Index 0 Level 0 Class name voice_prec6 service_id 0x0 Policy name voice_flow
Node flags: LEAF
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 5000kbps/62500B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102DD
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001803/0x005102DD/0x005102DE/0x005102DF
----------------------------------------------------
Index 1 Level 0 Class name ubrl1 service_id 0x0 Policy name voice_flow
Flow QoS Info: UBRL
UBRL Flow Mask Mode: src-ip (0x80)
UBRL Flow Age: 200 Seconds
Number of UBRL flows learnt: 0
Node flags: LEAF
Stats flags: Policer type 1 Max category 0
Node Config:
Flow Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 10000kbps/125000B/0kbps/0B
Catch-all Police Color aware 0 Type 1 CIR/CBS/PIR/PBS:
100000000kbps/12500000B/100000000kbps/12500000B
Node Result: Class-based stats:Stat ID 0x005102E0
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001804/0x005102E0/0x005102E1/0x005102E2
----------------------------------------------------
Index 2 Level 0 Class name class-default service_id 0x0 Policy name voice_flow
Node flags: LEAF DEFAULT DEFAULT-ALL
Stats flags: Policer type 1 Max category 0
Node Config:
Police Color aware 0 Type 1 CIR/CBS/PIR/PBS: 30000kbps/375000B/0kbps/0B
Node Result: Class-based stats:Stat ID 0x005102E3
Queue: N/A Stat ID(Commit/Excess/Drop): 0x00000000/0x00000000/0x00000000
Police ID (Token/Conform/Exceed/Violate): 0x00001805/0x005102E3/0x005102E4/0x005102E5
----------------------------------------------------
```

This is the sample output of the show qos-ea interface command for PWHE interface:

```
RP/0/# show qos-ea interface pw-ether 1 output member tenGigE 0/2/0/3
Thu Mar 20 01:46:39.070 UTC
Interface: TenGigE0_2_0_3 output policy: pwhe-egress
Total number of classes: 1
----------------------------------------------------
Policy name: pwhe-egress
Hierarchical depth 1
Interface type unknown
Interface rate 10000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x000000E0
ul_ifh 0x08000680, ul_id 0x00000040
uidb index 0x0019
```

```
qos_ifh 0x811a00019001e
Local port 3, NP 1
Policy map id 0x2000, format 15, uidb index 0x0019
------------------------------------------------------
Index 0 Level 0 Class name class-default service_id 0x0 Policy name pwhe-egress
Node flags: LEAF Q_LEAF DEFAULT DEFAULT-ALL
Stats flags: Queuing enabled
Node Config:
Shape: CIR/CBS/PIR/PBS: 0kbps/12500000B/1000000kbps/12500000B
WFQ: BW/Sum of BW/Excess ratio: 0kbps/0kbps/20
Queue limit 12500000 Guarantee 0
Node Result: Class-based stats:Stat ID 0x005102C1
Queue: Q-ID 0x0001000a Stat ID(Commit/Excess/Drop): 0x00640032/0x00000000/0x00640033
------------------------------------------------------
```

# show qos status interface

To display the status of the service-policy applied on the interface, use the **show qos status interface** command in EXEC mode.

**show qos status interface** [**interface** {*typeall*} *interface-path-id*] [**input** | **output**]**nv** [**satellite** *satellite-id*]

## Syntax Description

| | |
|---|---|
| *type* | Interface type. For more information, use the question mark (?) online help function. |
| *interface-path-id* | Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.<br>• *rack*—Chassis number of the rack.<br>• *slot*—Physical slot number of the line card.<br>• *module*—Module number. A physical layer interface module (PLIM) is always 0.<br>• *port*—Physical port number of the interface.<br>**Note**<br>In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 or RSP1) and the module is CPU0.<br>Example: interface MgmtEth0/RSP0/CPU0/0.<br>For more information about the syntax for the router, use the question mark (?) online help function. |
| **input** | (Optional) Displays per class statistics on inbound traffic for the specified policy map and interface. |
| **output** | (Optional) Displays per class statistics on outbound traffic for the specified policy map and interface. |
| **nv** | (Optional) Displays the statistics of the satellite access or fabric interfaces. |
| **satellite** | (Optional) Displays the per class statistics on the inbound or outbound traffic for the specified satellite. |
| *satellite-id* | (Optional) Displays the satellite statistics based on the specified satellite ID. |

## Command Default

No default behavior or values

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 5.1.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| qos | read, write |

This sample output shows how to the status of the service-policy applied on the TenGigE interface 0/6/1/0 that are in the output and input direction:

```
RP/0/RSP0/CPU0:router#  show qos status interface gigabitEthernet 200/0/0/0 nv

GigabitEthernet200/0/0/0  Satellite: 200 input: marking

     Last Operation Attempted :  ADD
     Status                   :  ACTIVE

GigabitEthernet200/0/0/0 direction output: Service Policy not installed

RP/0/RSP0/CPU0:router#show qos status interface gigabitEthernet 200/0/0/0 input nv

GigabitEthernet200/0/0/0  Satellite: 200 input: marking

     Last Operation Attempted :  ADD
     Status                   :  ACTIVE
```