Hierarchical Modular QoS Commands

This chapter provides details of the Hierarchical QoS commands.

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.

Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management. For detailed HQoS concepts, configuration tasks and examples, see the Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers

- exceed-color, on page 2
- fragment, on page 4
- match dei, on page 5
- set dei, on page 7
- service-fragment, on page 9
- service-fragment-parent, on page 10
exceed-color

To configure preclassification of Frame Relay packets that are discard-eligible, use the `exceed-color` command in policy map police configuration mode. To remove an exceed color from the policy-map, use the `no` form of this command.

```
exceed-color class-map-name
no exceed-color class-map-name
```

**Syntax Description**

`class-map-name` Specifies the class-map to associate with the exceed-color.

**Command Default**

By default, if no preclassification is configured for a packet, the packet is not analyzed by the color-aware policer on the ingress interface, and the packet is given regular policing treatment.

**Command Modes**

Policy map police configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.1.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Typically, frame relay packets from a previous node are marked by default as fr-de = 0 (meaning not discard eligible) or fr-de = 1 (meaning discard eligible). For discard-eligible treatment, you must create a class map for the fr-de=1 case and assign the exceed-color to that class-map.

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

**Note**

The multi-action policer sets cannot be used for IP packets.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>qos</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

In this example, exceed-color is configured for preclassification of packets that are discard-eligible.

```
RP/0/RSP0/CPU0:router(config)# class-map match-all match_frde
RP/0/RSP0/CPU0:router(config-cmap)# match fr-de 1
RP/0/RSP0/CPU0:router(config-cmap)# policy-map 2R3C_exceed_example
RP/0/RSP0/CPU0:router(config-pmap)# class class-default
RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 768000 burst 288000 peak-rate 1536000 peak-burst 576000
RP/0/RSP0/CPU0:router(config-pmap-c-policer)# exceed-color match_frde
RP/0/RSP0/CPU0:router(config-pmap-c-policer)# exceed-action set qos-group 2
RP/0/RSP0/CPU0:router(config-pmap-c-policer)# exit
RP/0/RSP0/CPU0:router(config-pmap-c-policer)# exit
```
RP/0/RSP0/CPU0:router(config-pmap)# exit
RP/0/RSP0/CPU0:router(config)# interface pos 0/2/0/0
RP/0/RSP0/CPU0:router(config-if)# service-policy input policy2

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>child-conform-aware</td>
<td>Prevents the parent policer from dropping any ingress traffic that conforms to the maximum rate specified in the child policer.</td>
</tr>
<tr>
<td>class-map</td>
<td>Defines a traffic class and the associated rules that match packets to the class.</td>
</tr>
<tr>
<td>conform-action</td>
<td>Configures the action to take on packets that conform to the rate limit.</td>
</tr>
<tr>
<td>conform-color</td>
<td>(Used for SIP 700 cards only.) Configures preclassification of ingress Layer 2 Frame Relay packets that have been previously marked as not discard eligible on an upstream node. These previously-marked packets are analyzed and preclassified by the color-aware policer on the ingress interface as part of the 2-rate 3-color (2R3C) traffic policing feature.</td>
</tr>
<tr>
<td>exceed-action</td>
<td>Configures the action to take on packets that exceed the rate limit.</td>
</tr>
<tr>
<td>match fr-de</td>
<td>Match packets on the basis of the Frame Relay discard eligibility (DE) bit setting.</td>
</tr>
<tr>
<td>police rate</td>
<td>Configures traffic policing and enters policy map police configuration mode.</td>
</tr>
<tr>
<td>policy-map</td>
<td>Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.</td>
</tr>
<tr>
<td>show policy-map interface</td>
<td>Displays policy configuration information for all classes configured for all service policies on the specified interface.</td>
</tr>
<tr>
<td>violate-action</td>
<td>Configures the action to take on packets that violate the rate limit.</td>
</tr>
</tbody>
</table>
To refer to a defined service fragment, use the `fragment` command in the policy-map configuration mode. To delete an earlier mention of the service fragment, use the `no` form of the command.

```
fragment name
no fragment name
```

**Syntax Description**
- `name` Previously defined service-fragment.

**Command Default**
None

**Command Modes**
Policy-map configuration mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The number of defined service fragments on a port policy is the same as the number of classes supported in a policy.

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

This example shows how to refer to a previously-defined service fragment:

```
RP/0/RSP0/CPU0:router (config) # policy-map p1
RP/0/RSP0/CPU0:router (config-pmap) # class c1
RP/0/RSP0/CPU0:router (config-pmap-c) # fragment sf1
```
match dei

To specify a drop eligible indicator (DEI) value as a match criteria in a class map, use the `match dei` command in class map configuration mode. To remove a specified DEI value from the matching criteria for a class map, use the `no` form of this command.

```
match dei value
no match dei
```

**Syntax Description**

```
value    Value of the DEI bit. Can be 0 or 1.
```

**Command Default**

There is no default DEI value; it must be specified.

**Command Modes**

Class map configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7.3</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `match dei` command specifies a DEI 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.

**Examples**

In this example, 802.1ad CoS plus DEI is derived from the incoming 802.1q CoS. Packets with a CoS value of 0 are remarked with a DEI value of 1.

```
RP/0/RSP0/CPU0:router(config)# class-map match-any remark-cos
RP/0/RSP0/CPU0:router(config-cmap)# match cos 0
RP/0/RSP0/CPU0:router(config-cmap)# end-class-map
RP/0/RSP0/CPU0:router(config)# policy-map pl
RP/0/RSP0/CPU0:router(config-pmap-c)# class remark-cos
RP/0/RSP0/CPU0:router(config-pmap-c)# set dei 1
RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map

RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/4/0/39.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 1
RP/0/RSP0/CPU0:router(config-subif)# rewrite ingress tag push dot1ad 5 symmetric
RP/0/RSP0/CPU0:router(config-subif)# service-policy input pl
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class-map</td>
<td>Defines a traffic class and the associated rules that match packets to the class.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>match cos</td>
<td>Identifies specified class of service (CoS) values as a match criteria in a class map.</td>
</tr>
</tbody>
</table>
**set dei**

To set the drop eligible indicator (DEI) value in a policy map class, use the `set dei` command in policy map class configuration mode. To remove a specified DEI value from a policy map class, use the `no` form of this command.

```
set dei value
no set dei
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Value of the DEI bit. Can be 0 or 1.</td>
</tr>
</tbody>
</table>

**Command Default**

There is no default DEI value; it must be specified.

**Command Modes**

Policy map class configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `set dei` command specifies a DEI value in a policy map class. For example, traffic can be policed and the excess traffic can be marked with DEI value of 1, so that it can be preferentially dropped in the egress interface or further downstream, when there is congestion.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>qos</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

In this example, 802.1ad CoS plus DEI is derived from the incoming 802.1q CoS. Packets with a CoS value of 0 are remarked with a DEI value of 1.

```
RP/0/RSP0/CPU0:router(config)# class-map match-any remark-cos
RP/0/RSP0/CPU0:router(config-cmap)# match cos 0
RP/0/RSP0/CPU0:router(config-cmap)# end-class-map
RP/0/RSP0/CPU0:router(config)# policy-map pl
RP/0/RSP0/CPU0:router(config-pmap)# class remark-cos
RP/0/RSP0/CPU0:router(config-pmap-c)# set dei 1
RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map

RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/4/0/39.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 1
RP/0/RSP0/CPU0:router(config-subif)# rewrite ingress tag push dot1ad 5 symmetric
RP/0/RSP0/CPU0:router(config-subif)# service-policy input pl
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class (policy-map)</td>
<td>Specifies the name of the class whose policy you want to create or change.</td>
</tr>
</tbody>
</table>
### Hierarchical Modular QoS Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class-map</td>
<td>Defines a traffic class and the associated rules that match packets to the</td>
</tr>
<tr>
<td></td>
<td>class.</td>
</tr>
<tr>
<td>policy-map</td>
<td>Creates or modifies a policy map that can be attached to one or more</td>
</tr>
<tr>
<td></td>
<td>interfaces to specify a service policy.</td>
</tr>
</tbody>
</table>
service-fragment

To define a service fragment in a class, use the `service-fragment` command in the policy-map configuration mode. To delete the service-fragment, use the `no` form of this command.

```
service-fragment name
no service-fragment name
```

**Syntax Description**

- `name` Name for the service fragment.

**Command Default**

None

**Command Modes**

Policy-map configuration mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Service fragment names must be unique in a port policy. However, same names can be used across policies.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>qos</td>
<td>read, write</td>
</tr>
</tbody>
</table>

This example shows how to define a service fragment name:

```
RP/0/RSP0/CPU0:router (config) # policy-map p1
RP/0/RSP0/CPU0:router (config-pmap) # class c1
RP/0/RSP0/CPU0:router (config-pmap-c) # service-fragment sf1
```
service-fragment-parent

To apply a service fragment policy to an interface, use the `service-fragment-parent` command in the interface configuration mode. To delete the applied service fragment policy, use the `no` form of this command.

```
service-fragment-parent
no service-fragment-parent
```

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
No default behavior or value

**Command Modes**
Interface configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Do not use this command was non-service fragment policies.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>qos</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to use the `service-fragment-parent` command:

```
RP/0/RSP0/CPU0:router (config-if) # service-policy input s1 service-fragment-parent
```