



APPENDIX **F**

CLI Command Reference for QPM Actions

QPM uses device commands to configure your QoS policies and configurations on the devices. These are the same commands you can use on the device's command line interface (CLI), and they are described in the device's documentation.

This section shows the command sequences used to configure each type of abstract policy action that you can create using QPM. You can use this information to help you understand how QPM configures your devices.

See the device's documentation for complete information on the commands and their parameters. (See [More Information About Quality of Service, page 2-30](#) for a partial list of product documentation.)

These sections show the full command translation, including optional parameters. If you do not configure an optional setting, the associated command or parameter is not included in the command sequence QPM uses to configure the device.



Note

The notation in the translation is **bold** for the device's key words, *italic* for variables. Some of the variables are parameters you enter into QPM. Other variables are managed by QPM, for example, the ACL number.

These sections describe QPM abstract actions:

- [Time Based ACL, page F-4](#)
- [Access List Configuration, page F-4](#)
- [Named ACL, page F-5](#)

- [Class-Based QoS Configuration](#), page F-5
- [Class-Based QoS Marking](#), page F-7
- [Class-Based QoS Policing](#), page F-8
- [Class-Based QoS Shaping](#), page F-8
- [Modular Shaping](#), page F-9
- [FIFO Queuing Configuration](#), page F-9
- [WFQ Configuration](#), page F-9
- [WFQ on VIP Cards \(DWFQ with QoS Group\) Configuration](#), page F-10
- [FRTS Configuration](#), page F-10
- [WFQ with FRTS Configuration](#), page F-11
- [FRTS with FRF.12 \(Voice Configuration\) Configuration](#), page F-12
- [WRED Configuration](#), page F-12
- [Priority Queuing Configuration](#), page F-13
- [Custom Queuing Configuration](#), page F-13
- [Shaped Round-Robin \(SRR\) Policies](#), page F-14
- [Weighted Round-Robin \(WRR\) Policies](#), page F-15
- [NBAR Port Map Configuration](#), page F-15
- [NBAR Custom Map Configuration](#), page F-15
- [PVC Bundle Configuration](#), page F-16
- [RSVP Configuration](#), page F-16
- [IP RTP Priority Configuration](#), page F-16
- [CRTP Configuration](#), page F-17
- [LFI Configuration](#), page F-17
- [TX-Ring Configuration](#), page F-17
- [Inline Power](#), page F-18
- [QoS Pre-classification](#), page F-18
- [CoS Configuration](#), page F-18
- [Maximum Reserved Bandwidth](#), page F-18
- [Access Control Policies](#), page F-19

- Router Marking Policies (PBR), page F-19
- Policing Policies (CAR), page F-19
- Shaping Policies (GTS), page F-20
- RMON Alarms, page F-20
- RMON Events, page F-20
- Catalyst 2900XL and Catalyst 3500XL Marking Policies, page F-21
- Catalyst 2980, 4000, 4500 Queuing Policies, page F-21
- Catalyst 2980, 4000, 4500 Marking Policies, page F-21
- Catalyst 5000 Marking Policies, page F-21
- Catalyst 6000 2Q2T and 1P2Q2T Queuing Configuration, page F-22
- Catalyst 6000 CoS, Precedence, DSCP, and DSCP Markdown Mapping, page F-22
- Catalyst 6000 Port Configuration, page F-23
- Catalyst 6000 Marking Policies, page F-23
- Catalyst 6000 Policing Policies, page F-23
- Configuration on Catalyst Switches with Supervisor IOS Software, page F-24
- Layer 3 Policing Policies, page F-33
- Layer 3 Shaping Policies, page F-33
- AutoQoS Configuration, page F-33

Time Based ACL

When you configure time-of-day Quality of Service (QoS) policies in a router running Cisco IOS software, QPM uses this command sequence:

1. **time-range** *time-range-name*
2. Either
 - a. **absolute** [*start time date*] [*end time date*]
 Or
 - b. **periodic** [*days-of-the-week*] *hh:mm to* [*days-of-the-week*] *hh:mm*

Access List Configuration

When you create filters for non class-based policies, QPM translates the filter definitions to ACLs using this command sequence:

- **access-list** *acl-index* {**deny** | **permit**} *protocol source source-wildcard* [{**eq** *src-port* | **range** *src-port-from src-port-to*}] *destination destination-wildcard* [{**eq** *dest-port* | **range** *dest-port-from dest-port-to*}] [**precedence** *precedence*] [**dscp** *dscp*]
- **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]

Named ACL

When you create filters for class-based policies, QPM uses this command sequence to configure ACLs on the device, if it supports filter names:

1. **ip access-list extended** *name*
 - **deny** | **permit** *protocol source source-wildcard* [{**eq** *src-port* | **range** *src-port-from src-port-to*}] *destination destination-wildcard* [{**eq** *dest-port* | **range** *dest-port-from dest-port-to*}] [**precedence** *precedence*] [**dscp** *dscp*]
 - **deny** | **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]

Class-Based QoS Configuration

When you select Class Based QoS as a QoS property, and create class-based queuing policies on the interface, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum filter*
2. **class-map** [**match all** | **match any**] *classname*
 - a. **match** [**not**] **access-group** *ACLNum*
 - b. **match** [**not**] **ip dscp** *dscp*
 - c. **match** [**not**] **ip precedence** *precedence*
 - d. **match** [**not**] **cos** *cos*
 - e. **match** [**not**] **ip rtp** *low_port range*
 - f. **match** [**not**] **protocol** *protocol [parametername [value]]*
 - g. **match** [**not**] **class-map** *classname*
 - h. **match** [**not**] **mpls experimental** *value*

3. **policy-map** *policy-map-name*

a. **class** { *classname* | **class-default** }

set ip precedence *precedence*

set ip dscp *dscp*

set cos *cos-value*

set fr-de

set mpls experimental *value*

police *police-rate* [*police-bc*] [**pir** *pir*] [**be** *police-be*] **conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]

police cir percent *percent* [**bc** *conform-burst-in-msec*] [**pir** *percent percent*] [**be** *peak-burst-in-msec*]

where *action* is:

{ **transmit** | **continue** | **set-prec-transmit** *precedence* |
set-dscp-transmit *dscp* | **set-prec-continue** *precedence* |
set-dscp-continue *dscp* | **drop** }

shape average *shape-rate* [*shape-bc shape-be*]

shape peak *shape-rate* [*shape-bc shape-be*]

shape adaptive *shape-adaptive-rate*

shape fecn-adapt

bandwidth *bandwidth*

bandwidth percent *percent*

bandwidth remaining percent *percent*

priority *bandwidth* [*burst*]

priority percent *percent* [*burst*]

fair queue *number-of-queues*

fair queue queue-limit *individual-queue-limit*

queue-limit *queue-limit*

random-detect (see [WRED Configuration, page F-12](#) for the random-detect commands)

4. **interface** *interfacename*
 - a. **service-policy** *direction policy-map-name*

**Note**

Some commands are for class-based QoS on a device that supports NBAR or IP RTP.

On ATM VCs, this command sequence is used:

1. **interface** *interfacename*
2. **pvc** *pvc-name*
 - a. **service-policy** *direction policy-map-name*

On frame-relay interfaces, this command sequence is used:

1. **map-class frame-relay** *classname*
 - a. **service-policy** [**input** | **output**] *policyname*
2. **interface** *interfacename*
 - a. **frame-relay class** *classname*

Class-Based QoS Marking

When you select Class Based QoS as a QoS property, and create marking policies on the interface, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
 - a. **class** *classname*
 - set ip precedence** *precedence*
 - set ip dscp** *dscp*
 - set cos** *cos-value*
 - set fr-de**
 - set mpls experimental** *value*

Class-Based QoS Policing

When you select Class Based QoS as a QoS property, and create policing policies on the interface, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
 - a. **class** *classname*

```
police police-rate [police-bc] [pir pir] [be police-be] conform-action
action [exceed-action action [violate-action action]]
```

where *action* is:

```
{ transmit | continue | set-prec-transmit precedence |
set-dscp-transmit dscp | set-prec-continue precedence |
set-dscp-continue dscp | drop }
```

Class-Based QoS Shaping

When you select Class Based QoS as a QoS property, and create shaping policies on the interface, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
 - a. **class** *classname*

```
shape average shape-rate [shape-bc shape-be]
shape peak shape-rate [shape-bc shape-be]
shape adaptive shape-adaptive-rate
shape fecn-adapt
```

Modular Shaping

When you select modular shaping as a QoS property, and create shaping policies, QPM uses this command sequence to configure the device:

1. **policy-map** *out_policies*
CLI out_policies configuration
2. **policy-map** *policy-map-name*
class class-default
 - a. **shape** {**average** | **peak**} *shape-rate* [*shape-bc shape-be*]
shape adaptive *shape-adaptive-rate*
shape fecn-adapt
 - b. **service-policy** *out_policies*
3. **interface** *interfacename*
service-policy output *policy-map-name*

FIFO Queuing Configuration

When you select FIFO as a QoS property, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **no fair-queue**

WFQ Configuration

When you select WFQ as a QoS property, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **fair-queue**

WFQ on VIP Cards (DWFQ with QoS Group) Configuration

When you select WFQ or FQ as a QoS property, and that interface is on a VIP card, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum condition*
2. **rate-limit output access-group** *ACLNum rate bc be conform-action transmit exceed-action drop*
3. **interface** *interfacename*
4. **fair-queue qos-group**
5. **fair-queue qos-group** *qos-group weight weight*
6. **fair-queue qos-group** *qos-group limit limit*
7. **fair-queue aggregate-limit** *aggregate-packet*
8. **fair-queue individual-limit** *individual-packet*

FRTS Configuration

When you enable Frame Relay traffic shaping (FRTS) on an interface, QPM uses this command sequence to configure the device:

1. **map-class frame-relay** *classname*
2. **frame-relay cir** *cir*
3. **frame-relay mincir** *mincir*
4. **frame-relay bc** *bc*
5. **frame-relay be** *be*
6. **frame-relay traffic-rate** *average [peak]*
7. **frame-relay adaptive-shaping** { *becn | foresight* }
8. **frame-relay ip rtp priority** *low range bandwidth*
9. **no frame-relay adaptive shaping**
10. **no frame-relay** { *adaptive-shaping becn | adaptive-shaping foresight* }

11. **interface** *interfacename*
 - a. **frame-relay traffic-shaping**
 - b. **frame-relay class** *classname*
 - c. **frame-relay ip rtp header-compression** [passive]

**Note**

If you are using FRTS with different queuing types, additional commands are available. See the relevant queuing commands for information about these commands.

If you are using FRTS on DLCI, the following commands are used:

- **interface** *interfacename*
 - **frame-relay traffic-shaping**
 - **frame-relay interface-dlci** *dlci-name*
 - **class** *classname*

WFQ with FRTS Configuration

When you select WFQ as a QoS property, and you enable Frame Relay traffic shaping (FRTS) on an interface, QPM uses this command sequence to configure the device:

1. **map-class frame-relay** *classname*
2. **frame-relay fair-queue** *congestive-discard-threshold*
number-dynamic-conversation-queues
number-reservable-conversation-queues max-buffer-size-for-fair-queues
3. **interface** *interfacename*
4. **frame-relay traffic-shaping**
5. **frame-relay class** *classname*

FRTS with FRF.12 (Voice Configuration) Configuration

When you enable Frame Relay traffic shaping (FRTS) on an interface, and configure the voice fields, QPM uses this command sequence to configure the device:

1. **map-class frame-relay** *classname*
2. **frame-relay fragment** *fragment-size*
3. **frame-relay voice bandwidth** *bps-reserved*
4. **interface** *interfacename*
5. **frame-relay traffic-shaping**
6. **frame-relay class** *classname*

WRED Configuration

When you select WRED as a QoS property, or select WRED for the drop mechanism for a class-based policy or interface QoS property, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **random-detect** *weight*

When you use advanced WRED the following commands are also available:

- **random-detect**
- **random-detect exponential-weighting-constant** *weight*
- **random-detect precedence** {*precedence* | **rsvp**} *min-threshold max-threshold probability-denominator*
- **random-detect dscp-based**
- **random-detect dscp** *dscp-value min-threshold max-threshold mark-probability-denominator*

Priority Queuing Configuration

When you select Priority Queuing as a QoS property, and create priority queuing policies on the interface, QPM uses this command sequence to configure the device (except for Frame Relay interfaces on which you have enabled FRTS):

1. **access-list** *ACLNum filter*
2. **priority-list** *priorityNum protocol ip level list ACLNum*
3. **priority-list** *priorityNum default level*
4. **priority-list** *priorityNum queue-limit high-limit medium-limit normal-limit low-limit*
5. **interface** *interfacename*
6. **priority-group** *priorityNum*

If the interface is Frame Relay using FRTS, QPM uses this command sequence to configure the device:

1. **map-class frame-relay** *classname*
2. **frame-relay priority-group** *priorityNum*
3. **interface** *interfacename*
4. **frame-relay class** *classname*

Custom Queuing Configuration

When you select Custom Queuing as a QoS property, and create custom queuing policies on the interface, QPM uses this command sequence to configure the device (except for Frame Relay interfaces on which you have enabled FRTS):

1. **access-list** *ACLNum filter*
2. **queue-list** *qListNum protocol ip qNum list ACLNum*
3. **queue-list** *qListNum queue qNum byte-count bytes [limit limit]*
4. **queue-list** *qListNum default qNum*
5. **interface** *interfacename*
6. **custom-queue-list** *qListNum*

If the interface is Frame Relay using FRTS, QPM uses this command sequence to configure the device:

1. **map-class frame-relay** *classname*
2. **frame-relay custom-queue-list** *qListNum*
3. **interface** *interfacename*
4. **frame-relay class** *classname*

Shaped Round-Robin (SRR) Policies

When you configure policies with Ingress Queue SRR configuration, QPM uses the following command sequence:

1. **mls qos srr-queue input cos-map queue** *queue-id* **threshold** *threshold-id* *cos1...cos8*
2. **mls qos srr-queue input threshold** *queue-id* *threshold-percentage1* *threshold-percentage2*
3. **mls qos srr-queue input buffers** *percentage1* *percentage2*
4. **mls qos srr-queue input bandwidth** *weight1* *weight2*
5. **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight*

When you configure policies with Egress Queue SRR configuration, QPM uses the following command sequence:

1. **mls qos queue-set output** *qset-id* **buffers** *allocation1* ... *allocation4*
2. **mls qos queue-set output** *qset-id* **threshold** *queue-id* *drop-threshold1* *drop-threshold2* *reserved-threshold* *maximum-threshold*
3. **mls qos srr-queue output cos-map queue** *queue-id* **threshold** *threshold-id* *cos1...cos8*

When you limit the maximum output on a port to a percentage of the port speed, QPM uses the following command:

- **srr-queue bandwidth limit** *weight1*

When you enable bandwidth shaping on the four egress queues mapped to a port, QPM uses the following command:

- **srr-queue bandwidth shape** *weight1* *weight2* *weight3* *weight4*

When you enable bandwidth sharing on the four egress queues mapped to a port, QPM uses the following command:

- **srr-queue bandwidth share** *weight1 weight2 weight3 weight4*

When you enable the priority of the queue for bandwidth shaping or sharing, QPM uses the following command:

- **priority-queue out**

Weighted Round-Robin (WRR) Policies

When you create queue weight policies for a layer 3 switch, QPM uses this command sequence to configure the device:

1. **qos switching**
2. **qos mapping** [**source** *Fastethernet name*] [**destination** *Fastethernet name*] **precedence** *precedence* **wrr-weight** *weight*

NBAR Port Map Configuration

When you enable NBAR port mapping, QPM uses this command sequence to configure the device:

- **ip nbar port-map** *protocol* {**tcp** | **udp**} *portnumbers*

NBAR Custom Map Configuration

When you use custom NBAR protocols to classify and monitor additional static port applications, QPM uses the following command to configure the device:

- **ip nbar custom** *name* [*offset format value*] [*source|destination*] [**tcp** | **udp**] [**range** *start end* | *port-number*]

PVC Bundle Configuration

When you add an ATM virtual circuit (VC) bundle member, QPM uses the following command sequence to configure the device:

1. **bundle** *bundle-name*
2. **pvc-bundle** *pvc-name*

When you add a Frame-relay virtual circuit (VC) bundle member, QPM uses the following command sequence to configure the device:

1. **frame-relay vc-bundle** *vc-bundle-name*
2. **pvc** [*vc-name*]

RSVP Configuration

When you enable resource reservation protocol (RSVP), QPM uses this command sequence to configure the device:

1. **ip rsvp bandwidth** [*interface-kbps* [*single-flow-kbps*]]
2. **ip rsvp udp-multicast**

IP RTP Priority Configuration

When you enable IP RTP priority, QPM uses this command sequence to configure the device:

- **ip rtp priority** *start-port port-range bandwidth*

CRTP Configuration

When you enable CRTP (RTP header compression), QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **ip rtp header-compression** [**passive**]

If the interface is Frame Relay, QPM uses this command sequence to configure the device:

- **frame-relay ip rtp header-compression** [**passive**]

LFI Configuration

When you enable LFI, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **ppp multilink interleave**
3. **ppp multilink fragment-delay** *delay*

TX-Ring Configuration

When you enable a TX-ring limit on a PVC, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **pvc** *pvc-name*
3. **tx-ring-limit** *ring-limit*

Inline Power

When you enable inline power on a switch running Catalyst OS, QPM uses this command sequence to configure the device:

- **set port inlinepower** *ports-list* **auto**

When you enable inline power on a Catalyst 4000 switch running Cisco IOS software, QPM uses this command sequence to configure the device:

- **interface** *interfacename*
- **power inline auto**

QoS Pre-classification

When you enable QoS preclassification for IP packets in tunnel interfaces, QPM uses the following command:

- **qos pre-classify**

CoS Configuration

When you override the previously configured trust state of the incoming packets and apply the default port CoS value to all incoming packets, QPM uses the following command:

- **mls qos cos** {*default-cos* | **override**}

Maximum Reserved Bandwidth

When you change the 75 percent rule of the interface bandwidth (for fancy queuing) under PVC, by specifying the maximum reserved bandwidth, QPM uses the following command:

- **max-reserved-bandwidth** *bandwidth*

Access Control Policies

When you create access control policies, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum filter*
2. **interface** *interfacename*
3. **ip access-group** *ACLNum direction*

Router Marking Policies (PBR)

When you create marking policies, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum filter*
2. **route-map** *tag permit [sequence]*
3. **match ip address** *ACLNum*
4. **set ip precedence** *precedence*
5. **interface** *interfacename*
6. **ip policy route-map** *tag*

Policing Policies (CAR)

When you create policing policies, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum filter*
2. **interface** *interfacename*
3. **rate-limit** {**input** | **output**} **access-group** *ACLNum rate bc be conform-action* {**transmit** | **continue** | **set-prec-transmit** *precedence* | **set-dscp-transmit** *dscp* | **set-prec-continue** *precedence* | **set-dscp-continue** *dscp* | **drop**} **exceed-action** {**transmit** | **continue** | **set-prec-transmit** *precedence* | **set-dscp-transmit** *dscp* | **set-prec-continue** *precedence* | **set-dscp-continue** *dscp* | **drop**}

Shaping Policies (GTS)

When you create shaping policies, QPM uses this command sequence to configure the device:

1. **access-list** *ACLNum filter*
2. **interface** *interfacename*
3. **traffic-shape group** *ACLNum rate [bc be]*

RMON Alarms

When you set an RMON alarm on any **cbQoSClassMapStats** MIB object, using the 'delta' option, QPM uses the following command to configure the device:

- **rmon alarm** *alarmIndex alarmVariable alarmInterval delta rising-threshold alarmRisingThreshold [alarmRisingEventIndex] falling-threshold alarmFallingThreshold [alarmFallingEventIndex] [owner alarmOwner]*

RMON Events

When you set up an RMON event to trigger, once the RMON alarm threshold is violated, QPM uses the following command sequence to configure the device:

1. **rmon event** *eventIndex [trap eventCommunity] [description eventDescription] [owner eventOwner]*
2. **snmp-server community** *string [ro | rw]*
3. **snmp-server host** *<host-addr> notification [community <community>] [udp-port <port>]*
4. **snmp-server enable traps alarms**

Catalyst 2900XL and Catalyst 3500XL Marking Policies

When you configure Catalyst 2900XL and Catalyst 3500XL marking policies, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **Switchport priority default cos** *cos*
3. **Switchport priority extend** {*none* | *trust*}
4. **Switchport priority extend cos** *cos*

Catalyst 2980, 4000, 4500 Queuing Policies

When you configure 2Q1T queuing policies for Catalyst 2980, 4000, or 4500 switches, QPM uses this command sequence to configure the device:

- **set qos map** *queue-type qid threshold cos cos-value*

Catalyst 2980, 4000, 4500 Marking Policies

When you configure marking policies for Catalyst 2980, 4000, or 4500 switches, QPM uses this command sequence to configure the device:

- **set qos defaultcos** *cos*

Catalyst 5000 Marking Policies

When you create marking policies for a Catalyst 5000 family switch, QPM uses this command sequence to configure the device:

1. **set qos enable**
2. **set qos ip-filter** *precedence protocol source source-port destination destination-port*

Catalyst 6000 2Q2T and 1P2Q2T Queuing Configuration

When you configure 2Q2T and 1P2Q2T queuing for a Catalyst 6000 family switch, QPM uses this command sequence to configure the device:

1. **set qos enable**
2. **set qos map 1p2q2t | 2q2t tx** *queue-number threshold-number cos cos-list*
3. **set qos drop-threshold 2q2t tx queue** *queue-number threshold-1 threshold-2*
4. **set qos wrr 1p2q2t | 2q2t** *queue-weight-1 queue-weight-2*
5. **set qos txq-ratio 1p2q2t | 2q2t** *queue-ratio-1 queue-ratio-2 [queue-ratio-3]*
6. **set qos wred 1p2q2t tx queue** *queue-number threshold-1 threshold-2*

Catalyst 6000 CoS, Precedence, DSCP, and DSCP Markdown Mapping

When you configure these mapping settings for a Catalyst 6000, QPM uses this command sequence to configure the device:

1. **set qos cos-dscp-map** *dscp dscp dscp dscp dscp dscp dscp dscp*
2. **set qos dscp-cos-map** *dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp:cos*
3. **set qos ipprec-dscp-map** *dscp dscp dscp dscp dscp dscp dscp dscp*
4. **set qos policed-dscp-map** *dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp:dscp*
5. **set qos policed-dscp-map excess-rate** *dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp, dscp:dscp*

Catalyst 6000 Port Configuration

When you configure the trust state for a Catalyst 6000 family switch port, QPM uses this command sequence to configure the device:

1. **set port qos *module/port* trust {trust-cos | trust-ipprec | trust-dscp}**
2. **set port qos *module/port* trust-ext {trust-cos | untrusted}**
3. **set port qos *module/port* {port-based | vlan-based}**

Catalyst 6000 Marking Policies

When you create marking policies for a Catalyst 6000 family switch, QPM uses this command sequence to configure the device:

1. **set qos acl ip *acl-name* dscp {*dscp* | trust-cos | trust-ipprec | trust-dscp} protocol source [{eq *port* | range *port port*}] destination [{eq *port* | range *port port*}] [{precedence *precedence* | dscp-field *dscp*}]**
2. **commit qos acl *acl-name***
3. **set qos acl map *acl-name* {*module/port* | *vlan*}**

Catalyst 6000 Policing Policies

When you create policing policies for a Catalyst 6000 family switch, QPM uses this command sequence to configure the device:

1. **set qos policer {aggregate | microflow} *policer-name* rate *rate* [policed-dscp *erate rate* {policed-dscp | drop}] burst *burst* [{policed-dscp | drop}]**
2. **set qos acl [default-action] ip *acl-name* dscp {*dscp* | trust-cos | trust-ipprec | trust-dscp} {aggregate | microflow} *policer-name* [protocol source [{eq *port* | range *port port*}] destination [{eq *port* | range *port port*}]**
3. **commit qos acl *acl-name***
4. **set qos acl map *acl-name* {*module/port* | *vlan*}**

Configuration on Catalyst Switches with Supervisor IOS Software

- [Port Configuration on Catalyst Switches with Supervisor IOS Software, page F-24](#)
- [Marking Policies on Catalyst Switches with Supervisor IOS Software, page F-26](#)
- [Policing Policies on Catalyst Switches with Supervisor IOS Software, page F-28](#)
- [Queuing on Catalyst Switches with Supervisor IOS Software, page F-29](#)
- [CoS, Precedence, DSCP, and DSCP Markdown Mapping on Catalyst Switches with Supervisor IOS Software, page F-32](#)

Port Configuration on Catalyst Switches with Supervisor IOS Software

The command sequences used to configure ports on Catalyst switches with Supervisor IOS software differs depending on the Catalyst model.

Catalyst 6000

When you configure the trust state for Catalyst 6000 switches with Supervisor IOS, QPM uses this command sequence to configure the device:

1. `mls qos`
2. `interface interfacename`
3. `mls qos`
4. `mls qos trust { cos | ip-precedence | dscp | device cisco-phone }`
5. `mls qos vlan-based`

Catalyst 4000, 4500

When you configure the trust state for Catalyst 4000 and 4500 switches with Supervisor IOS software, QPM uses this command sequence to configure the device:

1. **qos**
2. **interface** *interfacename*
3. **qos**
4. **qos trust [dscp | cos]**
5. **qos vlan-based**

Catalyst 2950, Catalyst 2960

When you configure the trust state for Catalyst 2950 and 2960 switches with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **mls qos**
2. **interface** *interfacename*
3. **mls qos**
4. **mls qos trust { cos | dscp | device cisco-phone }**

Catalyst 3550, Catalyst 3750

When you configure the trust state for Catalyst 3550 or 3750 switches with Supervisor IOS (no VLAN-based QoS), QPM uses this command sequence to configure the device:

1. **mls qos**
2. **interface** *interfacename*
3. **mls qos**
4. **mls qos trust { cos | ip-precedence | dscp | device cisco-phone }**

Marking Policies on Catalyst Switches with Supervisor IOS Software

The command sequences used to configure marking policies on Catalyst switches with Supervisor IOS software differs depending on the Catalyst model.

Catalyst 6000

When you configure marking policies for a Catalyst 6000 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **class-map** [**match all** | **match any**] *classname*
 - a. **match** [**not**] **access-group** *ACLNum*
2. **policy-map** *policy-map-name*
class *classname*
trust { **cos** | **ip-precedence** | **dscp** }

Catalyst 4000, 4500

When you configure marking policies for a Catalyst 4000 and 4500 with Supervisor IOS software, QPM uses this command sequence to configure the device:

1. **class-map** [**match all** | **match any**] *classname*
 - a. **match** [**not**] **access-group** *ACLNum*
2. **policy-map** *policy-map-name*
class *classname*
trust { **cos** | **dscp** }
set ip { **dscp** | **precedence** }

Catalyst 2950, 2960, 3550, 3750

When you configure marking policies for a class-default filter for Catalyst 2950 or 2960 with Supervisor IOS, or Catalyst 3550 or 3750 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **Switchport priority extend trust**
2. **Switchport priority extend cos** *cos*

When you configure marking policies for a user-defined filter for a Catalyst 2950 or 2960 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **class-map** [**match all** | **match any**] *classname*
 - a. **match** [**not**] **access-group** *ACLNum*
2. **policy-map** *policy-map-name*
3. **class** *classname*
4. **set ip dscp** *dscp*

When you configure marking (policing) policies for a user-defined filter for a Catalyst 3550 or 3750 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **class-map** [**match all** | **match any**] *classname*
 - a. **match** [**not**] **access-group** *ACLNum*
2. **policy-map** *policy-map-name*
3. **class** *classname*
4. **set ip dscp** *dscp*
5. **set ip precedence** *precedence*
6. **trust** { **cos** | **ip-precedence** | **dscp** }

Policing Policies on Catalyst Switches with Supervisor IOS Software

The command sequences used to configure policing on Catalyst switches with Supervisor IOS software differs depending on the Catalyst model.

Catalyst 6000, 3550, 3750

When you configure policing policies for a Catalyst 6000 with Supervisor IOS or Catalyst 3550 or 3750 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
 class *classname*
 - a. **police** [**flow**] *rate bc be* [**pir** *pir*] **conform-action** {**set-prec-transmit** | **set-dscp-transmit** | **drop**} **exceed-action** {**transmit** | **policed-dscp-transmit** | **drop**} [**violate-action** {**transmit** | **policed-dscp-transmit** | **drop**}]
 - b. **police** **aggregate** *policer-name*

If you define an aggregate policing policy, the following command is used:

- **mls qos aggregate-policer** *policer-name rate bc be* [**pir** *pir*] **conform-action** {**set-prec-transmit** | **set-dscp-transmit** | **drop**} **exceed-action** {**transmit** | **policed-dscp-transmit** | **drop**} [**violate-action** {**transmit** | **policed-dscp-transmit** | **drop**}]

Catalyst 4000, 4500

When you configure policing policies for a Catalyst 4000 and 4500 with Supervisor IOS software, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
 class *classname*
 - a. **police** *rate burst* [[**conform-action** { **transmit** | **drop** }] [**exceed-action** { **transmit** | **drop** | **policed-dscp-transmit** }]]
 - b. **police** **aggregate** *aggregate-name*

If you define an aggregate policing policy, the following command is used:

- **qos aggregate-policer** *policer-name rate burst* [[**conform-action** { **transmit** | **drop** }] [**exceed-action** { **transmit** | **drop** | **policed-dscp-transmit** }]]

Catalyst 2950, 2960

When you configure policing policies for a Catalyst 2950 or 2960 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **policy-map** *policy-map-name*
class *classname*
 - a. **police** *rate bc* [**exceed-action** { **dscp** *dscp* | **drop** }]
 - b. **police** *aggregate policer-name*

If you define an aggregate policing policy, the following command is used:

- **mls qos aggregate-policer** *policer-name rate bc be* [*pir* *pir*] **conform-action** { **set-prec-transmit** | **set-dscp-transmit** | **drop** } **exceed-action** { **transmit** | **policed-dscp-transmit** | **drop** } [**violate-action** { **transmit** | **policed-dscp-transmit** | **drop** }]

Queuing on Catalyst Switches with Supervisor IOS Software

The command sequences used to configure queuing on Catalyst switches with Supervisor IOS software differs depending on the Catalyst model.

Catalyst 6000

When you configure 2Q2T, 1P2Q2T, 1P3Q8T, 1P3Q1T, or 1P7Q8T queuing for Catalyst 6000 switches with Supervisor IOS, QPM uses the following command sequence:

1. **mls qos**
2. **interface** *interfacename*
3. **mls qos**
4. **wrr-queue cos-map** *queue-number threshold-number cos-list*
5. **priority-queue cos-map** *queue-id cos-list*

6. Any of these commands (to set the transmit-queue size ratio):
 - a. For 2Q2T or 1P2Q2T:
wrr-queue queue-limit *queue-ratio-1 queue-ratio-2*
 - b. For 1P3Q8T or 1P3Q1T:
wrr-queue queue-limit *queue-ratio-1 queue-ratio-2 queue-ratio-3*
 - c. For 1P7Q8T:
wrr-queue queue-limit *queue-ratio-1 queue-ratio-2 queue-ratio-3 queue-ratio-4 queue-ratio-5 queue-ratio-6 queue-ratio-7*
7. Any of these commands (to allocate bandwidth):
 - a. For 2Q2T or 1P2Q2T:
wrr-queue bandwidth *queue-weight-1 queue-weight-2*
 - b. For 1P3Q8T or 1P3Q1T:
wrr-queue bandwidth *queue-weight-1 queue-weight-2 queue-weight-3*
 - c. For 1P7Q8T:
wrr-queue bandwidth *queue-weight-1 queue-weight-2 queue-weight-3 queue-weight-4 queue-weight-5 queue-weight-6 queue-weight-7*
8. Any of these commands (to configure the drop-threshold):
 - a. For 1P3Q1T:
wrr-queue threshold *queue-number threshold-1*
 - b. For 2Q2T or 1P2Q2T:
wrr-queue threshold *queue-number threshold-1 threshold-2*
 - c. For 1P3Q8T or 1P7Q8T:
wrr-queue threshold *queue-number threshold-1 threshold-2 threshold-3 threshold-4 threshold-5 threshold-6 threshold-7 threshold-8*
9. Any of these commands (to specify the min and max WRED threshold):
 - a. For 1P3Q1T:
wrr-queue random-detect {max-threshold | min-threshold}
queue-number threshold-1
 - b. For 2Q2T or 1P2Q2T:
wrr-queue random-detect {max-threshold | min-threshold}
queue-number threshold-1 threshold-2
 - c. For 1P3Q8T or 1P7Q8T:
wrr-queue random-detect {max-threshold | min-threshold}
queue-number threshold-1 threshold-2 threshold-3 threshold-4 threshold-5 threshold-6 threshold-7 threshold-8

Catalyst 4000, 4500

When you configure 4Q1T Shape queuing for Catalyst 4000 and 4500 switches with Supervisor IOS software, QPM uses the following command sequence:

1. **qos map dscp** *dscp-values* **to tx-queue** *queue-id*
2. **tx-queue** *queue-id*
3. **bandwidth** *rate*
4. **shape** *rate*
5. **priority high**

Catalyst 3550

When you configure 4Q2T queuing for Catalyst 3550 switches with Supervisor IOS, QPM uses the following command sequence:

1. **wrr-queue cos-map** *queue-number cos-list*
2. **priority-queue out**
3. **wrr-queue dscp-map** *threshold-id dscp-list*
4. **wrr-queue queue-limit** *queue-ratio-1 queue-ratio-2 queue-ratio-3 queue-ratio-4*
5. **wrr-queue bandwidth** *queue-weight-1 queue-weight-2 queue-weight-3 queue-weight-4*
6. **wrr-queue threshold** *queue-number threshold-1 threshold-2*
7. **wrr-queue random-detect max-threshold** *queue-number threshold-1 threshold-2*

Catalyst 2950, 2960

When you configure 4Q1T queuing for Catalyst 2950 or 2960 switches with Supervisor IOS, QPM uses the following command sequence:

1. **wrr-queue bandwidth** *queue-weight-1 queue-weight-2 queue-weight-3 queue-weight-4*
2. **wrr-queue cos-map** *queue-number cos-list*

CoS, Precedence, DSCP, and DSCP Markdown Mapping on Catalyst Switches with Supervisor IOS Software

The command sequences used to configure these mappings on Catalyst switches with Supervisor IOS software differs depending on the Catalyst model.

Catalyst 6000, 3550, 3750

When you configure these mapping settings for a Catalyst 6000, 3550, or 3750, with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **mls qos map cos-dscp** *dscp dscp dscp dscp dscp dscp dscp*
2. **mls qos map dscp-cos** *dscp dscp dscp dscp dscp dscp dscp dscp* **to** *cos*
3. **mls qos map ip-prec-dscp** *dscp dscp dscp dscp dscp dscp dscp dscp*
4. **mls qos map policed-dscp** **[{normal-burst | max-burst}]** *dscp dscp dscp dscp dscp dscp dscp dscp* **to** *dscp*

Catalyst 4000, 4500

When you configure these mapping settings for a Catalyst 4000 and 4500 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **qos map cos** *cos-list* **to** **dscp** *dscp*
2. **qos map dscp policed** *dscp-list* **to** **dscp** *mark-down-dscp*
3. **qos map dscp** *dscp-list* **to** **cos** *cos*

Catalyst 2950, 2960

When you configure these mapping settings for a Catalyst 2950 or 2960 with Supervisor IOS, QPM uses this command sequence to configure the device:

1. **mls qos map cos-dscp** *dscp dscp dscp dscp dscp dscp dscp dscp*
2. **mls qos map dscp-cos** *dscp dscp dscp dscp dscp dscp dscp dscp* **to** *cos*

Layer 3 Policing Policies

When you configure Layer 3 policing policies, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **rate-limit** {**input** | **output**} *rate* [*burst*]

Layer 3 Shaping Policies

When you configure Layer 3 shaping policies, QPM uses this command sequence to configure the device:

1. **interface** *interfacename*
2. **traffic-shape rate** *rate* [*burst*]

AutoQoS Configuration

When you use AutoQoS for configuring, QPM uses these comands to configure the device:

- [Configuration for Cisco Catalyst 6500, page F-34](#)
- [Configuration for Cisco Catalyst 3550 and 2950EI, page F-34](#)
- [Configuration for Cisco Catalyst 4500 Switch, page F-34](#)
- [Configuration for Cisco Routers, page F-35](#)

Configuration for Cisco Catalyst 6500

When you configure AutoQoS on Cisco Catalyst 6500 Switches, QPM uses these commands to configure the device:

- **set qos autoqos**

To apply the automatic QoS feature on a per-port basis, use:

- **set port qos *mod/port* autoqos trust {cos | dscp}**
- **set port qos *mod/port* autoqos voip {ciscosoftphone | ciscoipphone}**

Configuration for Cisco Catalyst 3550 and 2950EI

When you configure AutoQoS on Cisco Cisco Catalyst 3550 and 2950EI Switches, QPM uses these commands to configure the device:

- **auto qos voip {cisco-phone | trust}**
 - **trust** — To configure auto-QoS for voice over IP (VoIP) within a QoS domain. QoS Labels in ingress packets are trusted.
 - **cisco-phone** — To Identify this interface as connected to a Cisco IP phone, and automatically configure QoS for VoIP. The QoS labels of incoming packets are trusted only when the phone is detected. Extends trust boundary if IP Phone detected.

Configuration for Cisco Catalyst 4500 Switch

When you configure AutoQoS on Cisco Cisco Catalyst 4500 Switches, QPM uses these commands to configure the device:

- **auto qos voip {cisco-phone | trust}**
 - **trust** — To configure auto-QoS for voice over IP (VoIP) within a QoS domain. QoS Labels in ingress packets are trusted.
 - **cisco-phone** — To Identify this interface as connected to a Cisco IP phone, and automatically configure QoS for VoIP. The QoS labels of incoming packets are trusted only when the phone is detected. Extends trust boundary if IP Phone detected.

Configuration for Cisco Routers

When you configure AutoQoS on Cisco Routers, QPM uses these commands to configure the device:

To configure the AutoQoS — VoIP feature on an interface:

- **auto qos voip** [**trust** | **fr-atm**]

To begin discovering and collecting data for configuring the AutoQoS for the Enterprise feature:

- **auto discovery qos** [**trust**]

(Untrusted Mode by default)

To install the QoS class maps and policy maps created by the AutoQoS:

- **auto qos**

