Using Modular QoS CLI

This chapter describes how to configure Modular QoS CLI (MQC) objects that can be used for configuring QoS features using the Cisco NX-OS software.

- Finding Feature Information, on page 1
- Information About MQC, on page 1
- Using an MQC Object, on page 2
- Configuring an MQC Object, on page 10
- Applying Descriptions to MQC Objects, on page 14
- Verifying an MQC Object, on page 15
- Attaching and Detaching a QoS Policy Action, on page 15
- Session Manager Support for QoS, on page 18
- Feature History for Using Modular QoS CLI, on page 19

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Information About MQC

MQC provides a language to define QoS policies.

For more information about MQC commands, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference*.

You configure QoS policies by following these three steps:

1. Define traffic classes.
2. Associate policies and actions with each traffic class.
3. Attach policies to logical or physical interfaces and VLANs.

MQC provides three command types to define traffic classes and policies:

- class-map—Defines a class map that represents a class of traffic based on packet-matching criteria. Class maps are referenced in policy maps.

Note When you configure match all for a QoS class map by entering the **class-map type qos match-all** command, the match-all option does not work. Instead, the match criteria is always treated as match any.

- table-map—Defines a table map that represents a mapping from one set of packet field values to another set of packet fields. Table maps are referenced in policy maps.
- policy-map—Defines a policy map that represents a set of policies to be applied on a class-by-class basis to class maps.

You define the following class-map and policy-map object types when you create them:

- network qos—Defines the characteristics of CoS properties network wide (across switches and VDCs).
- qos—Defines MQC objects that you can use for marking and policing.
- queuing—Defines MQC objects that you can use for queuing and scheduling.

Note The qos type is the default.

You can attach policies to ports, port channels, VLANs, subinterfaces, or tunnels by using the **service-policy** command.

On Fabric Extender (FEX) interfaces, you can configure only the type qos policies. However, you cannot configure the type qos policies that refer to classes that match with the access control lists (ACLs) that are configured for the FEX external interfaces.

The type queuing policies are currently not supported on FEX interfaces.

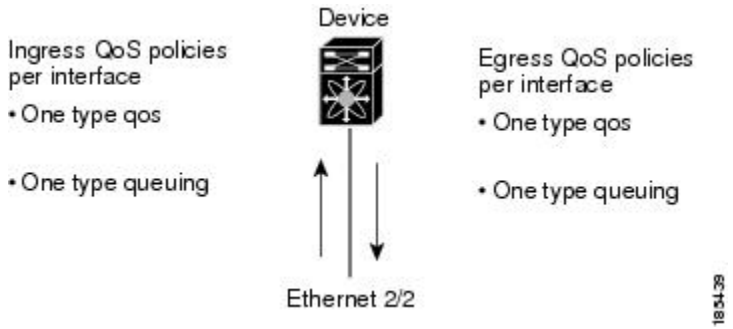
You can view all or individual values for MQC objects by using the **show table-map**, **show class-map**, and **show policy-map** commands.

Caution In interface configuration mode, the Cisco Nexus 7000 Series switch might accept QoS and ACL commands irrespective of the line card on which the interface host is up or down. However, you cannot enter interface submode when the line card is down because the Cisco Nexus 7000 Series switch does not accept any preconfiguration information.

Using an MQC Object

You configure QoS and queuing policies using the MQC class-map, policy-map, and table-map objects. You cannot use table maps in queuing policies. After you configure class maps and policy maps, you can attach one policy map of each type to each of the ingress or egress directions of an interface. The figure below lists the maximum QoS and queuing policies that you can define on each interface.

Figure 1: Maximum QoS Policies Per Interface



A policy map contains either a QoS policy or queuing policy. The policy map references the names of class maps that represent traffic classes. For each class of traffic, the device applies the policies on the interface or VLAN that you select.

A packet is matched sequentially to a class of traffic starting from the first traffic class definition. When a match is found, the policy actions for that class are applied to the packet.

The reserved class map receives all traffic that is not matched in type qos policies, and the device applies the policy actions as it would for any other traffic class. You use class-default to perform mutations (mutation is a method for translating QoS values in the packet header prior to traffic classification).

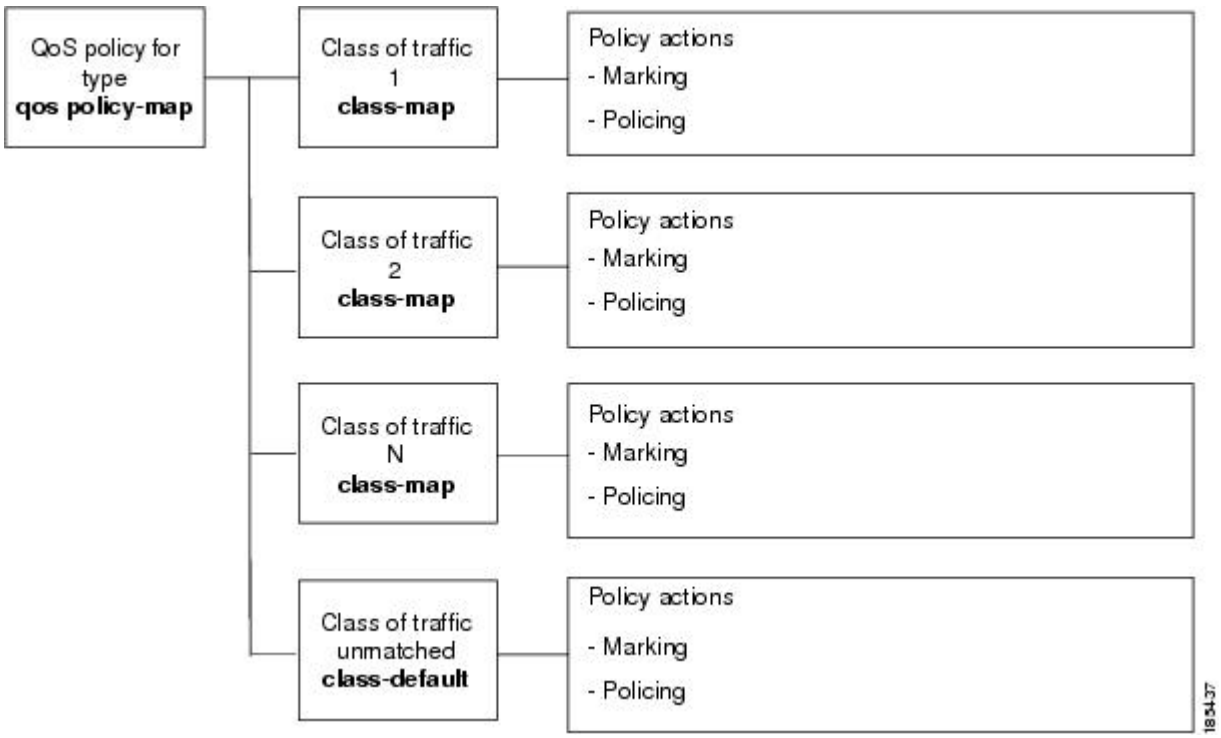
Note You can access user-defined MQC objects only in the VDC in which they were created. You can access the system-defined MQC objects in all VDCs.

Type qos Policies

You use type qos policies to mark, to apply mutations, to set the ingress port trust state, and to police packets.

The figure below shows the QoS policy structure with the associated MQC objects of type qos without mutation.

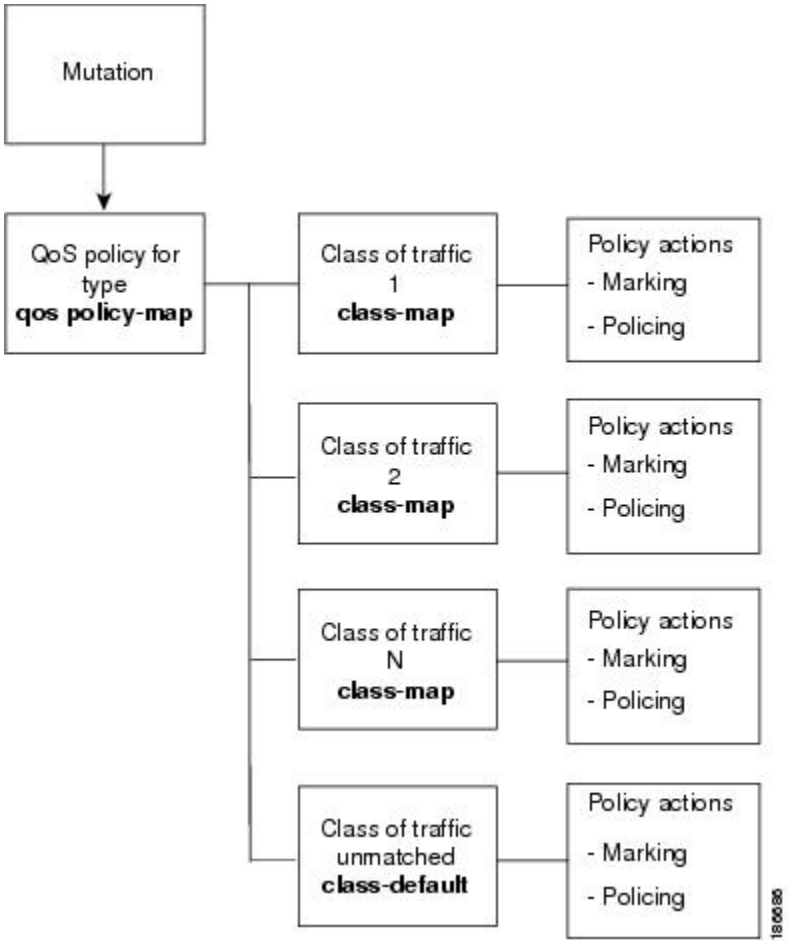
Figure 2: QoS Policy Diagram Showing Type qos MQC Object Usage Without Mutation



Note The MQC objects are shown in bold.

The figure below shows the QoS policy structure with mutation.

Figure 3: QoS Policy Diagram Showing Type qos MQC Object Usage with Mutation



Note The MQC objects are shown in bold.

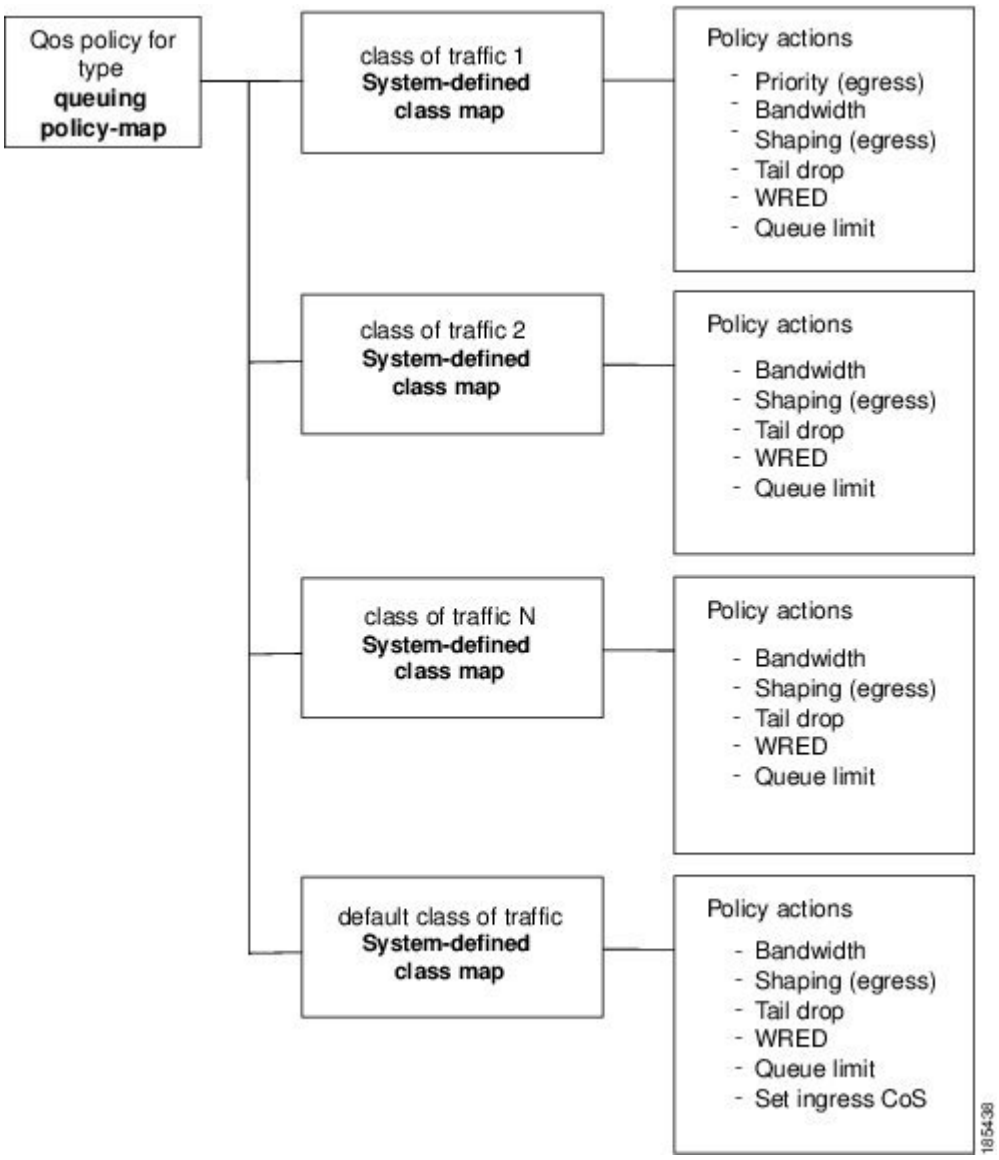
Type Queuing Policies

You use type queuing policies to mark, shape, and queue packets. Marking is limited to the CoS field and does not support the use of table maps.

The figure below shows the QoS policy structure with associated MQC objects of type queuing. The MQC objects are shown in bold.

Note MQC table-map objects cannot be used in policies of type queuing.

Figure 4: QoS Policy Diagram Showing Type Queuing MQC Object Usage



Note See "Queuing and Scheduling" for more information on configuring these parameters.

System-Defined MQC Objects

Note The system-defined MQC objects that are shown in the table below are the default. All of these values apply across all VDCs.

When you configure QoS features, and the system requests MQC objects, you can use one of the system-defined objects shown in the table below.

Table 1: System-Defined MQC Objects

Table	Description
System-Defined Type qos Class Maps	Type qos class maps
System-Defined Type queuing Class Maps	Type queuing class maps
System-Defined Table Maps	Table maps
System-Defined Queuing Policy Maps	Policy maps

Type qos class maps that are defined by the system are listed in the table below.

Note You cannot reference the conform-color-in, conform-color-out, exceed-color-in, or exceed-color-out class maps in a policy map.

Table 2: System-Defined Type qos Class Maps

Class Map Name	Description
class-default	Type qos class map that is assigned to all packets that match none of the criteria of traffic classes that you define in a type qos policy map. You can use class-default for mutation.
conform-color-in	Type qos conform color class map in the input direction. This color-aware class map makes a policer color-aware for a conform action.
conform-color-out	Type qos conform color class map in the output direction. This color-aware class map makes a policer color-aware for a conform action.
exceed-color-in	Type qos exceed color class map in the input direction. This color-aware class map makes a policer color-aware for an exceed action.
exceed-color-out	Type qos exceed color class map in the output direction. This color-aware class map makes a policer color-aware for an exceed action.

Type queuing class maps that are defined by the system are listed in the table below.

Table 3: System-Defined Type queuing Class Maps

Class Map Queue Name	Description	Default CoS Values
1 Gigabit Module Ingress: 2 queues with 4 thresholds per queue		

Class Map Queue Name	Description	Default CoS Values
2q4t-in-q1	Ingress queue 1 of 2q4t type	5-7
2q4t-in-q-default	Ingress default queue of 2q4t type	0-4
1 Gigabit Module Egress: 1 strict priority queue and 3 normal queues with 4 thresholds per queue		
1p3q4t-out-pq1 [1]	Egress priority queue of 1p3q4t type	5-7
1p3q4t-out-q2	Egress queue 2 of 1p3q4t type	—
1p3q4t-out-q3	Egress queue 3 of 1p3q4t type	—
1p3q4t-out-q-default	Egress default queue of 1p3q4t type	0-4
10 Gigabit Module Ingress: 8 queues with 2 thresholds per queue		
8q2t-in-q1	Ingress queue 1 of 8q2t type	5-7
8q2t-in-q2	Ingress queue 2 of 8q2t type	—
8q2t-in-q3	Ingress queue 3 of 8q2t type	—
8q2t-in-q4	Ingress queue 4 of 8q2t type	—
8q2t-in-q5	Ingress queue 5 of 8q2t type	—
8q2t-in-q6	Ingress queue 6 of 8q2t type	—
8q2t-in-q7	Ingress queue 7 of 8q2t type	—
8q2t-in-q-default	Ingress default queue of 8q2t type	0-4
10 Gigabit Module Egress: 1 strict priority queue and 7 normal queues with 4 thresholds per queue		
1p7q4t-out-pq1 [1]	Egress priority queue of 1p7q4t type	5-7
1p7q4t-out-q2	Egress queue 2 of 1p7q4t type	—
1p7q4t-out-q3	Egress queue 3 of 1p7q4t type	—
1p7q4t-out-q4	Egress queue 4 of 1p7q4t type	—
1p7q4t-out-q5	Egress queue 5 of 1p7q4t type	—
1p7q4t-out-q6	Egress queue 6 of 1p7q4t type	—
1p7q4t-out-q7	Egress queue 7 of 1p7q4t type	—
1p7q4t-out-q-default	Egress default queue of 1p7q4t type	0-4

[1] These are either priority or normal queues. If you use the priority keyword in your configuration, these queues are used as priority queues. Otherwise, they are used as normal queues.

Table maps that are defined by the system are listed in the table below. The default mapping of values in the tables maps is contained in RFC 2597. These table maps are not configurable.

Table 4: System-Defined Table Maps

Table Map Name	Description
cir-markdown-map	Table map used to mark down packets that exceed the committed information rate (CIR). **Note** Enter the **show table-map** command to display the default mapping.
pir-markdown-map	Table map used to mark down packets that violate the peak information rate (PIR). **Note** Enter the **show table-map** command to display the default mapping.
cos-discard-class-map	Table map used to map the CoS value to the discard-class value.
cos-dscp-map	Table map used to map the CoS value to the DSCP value.
cos-precedence-map	Table map used to map the CoS value to the precedence value.
dscp-cos-map	Table map used to map the DSCP value to the CoS value.
dscp-precedence-map	Table map used to map the DSCP value to the precedence value.
dscp-discard-class-map	Table map used to map the DSCP value to the discard-class value.
precedence-dscp-map	Table map used to map the precedence value to the DSCP value.
precedence-cos-map	Table map used to map the precedence value to the CoS value.
precedence-discard-class-map	Table map used to map the precedence value to the discard-class value.
discard-class-cos-map	Table map used to map the discard-class value to the CoS value.
discard-class-prec-map	Table map used to map the discard-class value to the precedence value.
discard-class-dscp-map	Table map used to map the discard-class value to the DSCP value.

Policy maps that are defined by the system are listed in the table below.

Table 5: System-Defined Queuing Policy Maps

Queuing Policy Map Name	Description
default-in-policy	Input queuing policy map that is attached to all module ports to which you do not apply a queuing policy map. The default configuration values are as follows: policy-map type queuing default-in-policy class type queuing in-q1 queue-limit percent 50 bandwidth percent 80 class type queuing in-q-default queue-limit percent 50 bandwidth percent 20
default-out-policy	Output queuing policy map that is attached to all module ports to which you do not apply a queuing policy map. The default configuration values are as follows: policy-map type queuing default-out-policy class type queuing out-pq1 priority level 1 queue-limit percent 16 class type queuing out-q2 queue-limit percent 1 class type queuing out-q3 queue-limit percent 1 class type queuing out-q-default queue-limit percent 82 bandwidth remaining percent 25

Configuring an MQC Object

When you specify an MQC object command, the device creates the object if it does not exist and then enters map mode.

To remove a class-map, table-map, or policy-map object, use the **no** form of the command that you used to create the object.

For the commands that you can use in the MQC object mode, see the following configuration chapters:

- Configuring Classification
- Configuring Marking
- Configuring Mutation Mapping
- Configuring Policing
- Configuring Queuing and Scheduling

Configuring or Modifying a Class Map

You can create or modify a class map. You can then reference class maps in policy maps.

Note You cannot create a queuing class map; you must use one of the system-defined queuing class maps listed in Table 3: System-Defined Type queuing Class Maps, on page 7

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name*	Creates or accesses the class map of type qos, and then enters class-map qos mode. Class-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters. **Note** When you configure match all for a QoS class map by entering the class-map type qos match-all command, the match-all option does not work. Instead, the match criteria is always treated as match any.
Step 3	switch(config-cmap-qos)# **exit**	Exits class-map qos mode and enters global configuration mode.
Step 4	(Optional) switch(config)# **class-map** [**type qos**] {**conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**}	Accesses the class map of type qos for one of the system-defined color maps, and then enters color-map mode. **Note** This command is only used when color-aware policing is required.
Step 5	switch(config-color-map)# **exit**	Exits color-map mode, and then enters global configuration mode.

	Command or Action	Purpose
Step 6	switch(config)# **class-map type queuing match-any** {*class-queuing-name* | *WORD*}	Creates or accesses the class map of type queuing, and then enters class-map queuing mode. Class queuing names are listed in Table 3: System-Defined Type queuing Class Maps, on page 7. **Note** The match on WORD is used for defining hierarchical class maps. The argument, WORD, is supported only on the F-Series Modules.
Step 7	switch(config-cmap-que)# **exit**	Exits class map queuing mode and enters global configuration mode.
Step 8	(Optional) switch(config)# **show class-map** [**type qos**] [*class-map-name* | **conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**]	Displays information about all configured class maps or a selected class map of type qos.
Step 9	(Optional) switch(config)# **show class-map type queuing** [*class-queuing-name*]	Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 3: System-Defined Type queuing Class Maps, on page 7.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Configuring or Modifying a Table Map

You can create or modify a table map that you can reference in policy maps. For information on configuring table maps, see “Configuring Marking.”

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **table-map** *table-map-name*	Creates or accesses the table map and then enters table-map mode. Table map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-tmap)# **exit**	Exits table-map mode and enters global configuration mode.

	Command or Action	Purpose
Step 4	switch(config)# **table-map** {**cir-markdown-map** | **pir-markdown-map**}	Accesses one of the system-defined markdown table maps, and then enters markdown-map mode.
Step 5	switch(config-mrkdwn-map)# **exit**	Exits table-map mode and enters global configuration mode.
Step 6	(Optional) switch(config)# **show table-map** [*table-map-name* | **cir-markdown-map** | **pir-markdown-map**}	Displays information about all configured table maps or a selected table map.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Configuring or Modifying a Policy Map

You can create or modify a policy map that you can use to define actions to perform on class maps.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **policy-map** [**type qos**] [**match-first**] {*qos-policy-map-name* | **qos-dynamic**}	Creates or accesses the policy map of type qos and then enters policy-map mode. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	switch(config-tmap)# **exit**	Exits policy-map mode and enters global configuration mode.
Step 4	switch(config)# **policy-map type queuing** [**match-first**] {*queuing-policy-map-name* | **qos-dynamic**}	Creates or accesses the policy map of type queuing and then enters policy-map mode. You can specify a policy-map name. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 5	switch(config-tmap)# **exit**	Exits policy-map mode and enters global configuration mode.
Step 6	(Optional) switch(config)# **show policy-map** [**type qos**] [*policy-map-name* | **qos-dynamic**]	Displays information about all configured policy maps or a selected policy map of type qos.
Step 7	(Optional) switch(config)# **show policy-map type queuing** [*policy-map-name* | **qos-dynamic**]	Displays information about all configured policy maps or a selected policy map of type queuing.

	Command or Action	Purpose
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Applying Descriptions to MQC Objects

You can use the **description** command to add a description to a MQC object.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**Option** / **Description** **class-map** [**type qos**] [**match-any** | **match-all**] *class-map-name* — Creates or accesses the class map, and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 alphanumeric characters. **table-map** *table-map-name* — Creates or accesses the table map, and then enters table-map mode. The table-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters **policy-map** [**type qos**] [**match-first**] {*qos-policy-map-name* | **qos-dynamic**} — Creates or accesses the policy map, and then enters policy-map mode. The policy-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.	

	Command or Action	Purpose
Step 3	switch(config-cmap)# **description** *string*	Adds a description string to the MQC object. The description can be up to 200 alphanumeric characters. **Note** You cannot modify the description of system-defined queuing class maps.
Step 4	switch(config-cmap)# **exit**	Exits table-map mode and enters global configuration mode.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Verifying an MQC Object

To display MQC object configuration information, perform one of the following tasks:

Command	Purpose
show class-map [**type qos**] [*class-map-name* | **conform-color-in** | **conform-color-out** | **exceed-color-in** | **exceed-color-out**]	Displays information about all configured class maps or a selected class map of type qos.
show class-map type queuing [*class-queuing-name*]	Displays information about all configured class maps or a selected class map of type queuing. Class queuing names are listed in Table 3: System-Defined Type queuing Class Maps, on page 7.
show table-map [*table-map-name* | **cir-markdown-map** | **pir-markdown-map**]	Displays information about all configured table maps or a selected table map.
show policy-map [**type qos**] [*policy-map-name* | **qos-dynamic**]	Displays information about all configured policy maps or a selected policy map of type qos.
show policy-map type queuing [*policy-map-name* | **qos-dynamic**]	Displays information about all configured policy maps or a selected policy map of type queuing.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference*.

Attaching and Detaching a QoS Policy Action

The software does not allow you to enable or disable QoS features with a configuration command. To enable or disable QoS features, you must attach or detach QoS policies to or from interfaces, VLANs, or tunnels as described in this section.

Note

- You must enable the tunnel feature by entering the **feature tunnel** command and configure the tunnel before you attach policies.
- On Fabric Extender (FEX) interfaces, you can configure only the type qos policies. However, you cannot configure the type qos policies that refer to classes that match with the access control lists (ACLs) that are configured for the FEX external interfaces.
- The type queuing policies are currently not supported on FEX interfaces.

The system-defined type queuing class maps (see Table 3: System-Defined Type queuing Class Maps, on page 7) are attached to each interface unless you specifically attach a different class map.

Note The device restricts QoS policies to one per interface per direction (ingress or egress) for each of the policy types qos and queuing.

Policies that are defined at multiple interfaces have the following restrictions:

- A QoS policy attached to the physical port takes effect when the port is not a member of a port channel.
- A QoS policy attached to a port channel takes effect even when policies are attached to member ports.
- A QoS policy attached to a VLAN is applied to all ports in that VLAN that do not have other policies specifically applied.
- One ingress policy type queuing is supported for each Layer 2 port- and Layer 2 port-channel interface in both the ingress and egress direction. Egress type qos policies are not allowed on Layer 2 port or Layer 2 port-channel interfaces.
- One ingress and one egress QoS policy are supported for each Layer 3 and Layer 3 port-channel interface.
- One ingress and one egress QoS policy are supported for each VLAN.
- One ingress and one egress queuing policy are supported for each Layer 2 port-, Layer 2 port-channel, Layer 3 port-, and Layer 3 port-channel interface.
- When a VLAN or port channel, or both, touches multiple forwarding engines, all policies that enforce a rate are enforced per forwarding engine.
- For example, if you configure a policer on a specific VLAN that limits the rate for the VLAN to 100 Mbps and if you configure one switch port in the VLAN on one module and another switch port in the VLAN on another module, each forwarding engine can enforce the 100-Mbps rate. In this case, you could actually have up to 200 Mbps in the VLAN that you configured to limit the rate to 100 Mbps.

Note Default queuing policies are active, unless you configure and apply another policy. For the default queuing policies, see Table 5: System-Defined Queuing Policy Maps, on page 10.

The interface where a QoS policy is applied is summarized in the table below. Each row represents the interface levels. The entry descriptions are as follows:

- Applied—Interface where an attached policy is applied.
- Present—Interface where a policy is attached but not applied.
- Not present—Interface where no policy is attached.
- Present or not—Interface where a policy is either attached or not, but not applied.

Port Policy	Port-Channel Policy	VLAN Policy
Applied	Not present	Present or not
Present or not	Applied	Present or not
Not present	Not present	Applied

To attach a policy map to an interface, tunnel, or VLAN, use the **service-policy** command. You can specify whether the policies defined in the policy map are applied to the input or output stream of packets on the interface.

To detach a policy map from an interface, tunnel, or VLAN, use the **no** form of the **service-policy** command.

Attaching a QoS Policy Action to an Interface or Tunnel

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {[**ethernet** *slot/port-list*] | [**tunnel** *number-list*]}	Enters interface mode on the Ethernet or tunnel interface. • *slot/port-list* is a space-separated list of slots and ports. • *number-list* is a space-separated list of tunnels.
Step 3	switch(config-if)# **service-policy** [**type qos**] {**input** | **output**} {*policy-map-name*} [**no-stats**]	Adds the policy map to the input or output packets of an interface. Only one input policy and one output policy can be attached to an interface.
Step 4	switch(config-if)# **exit**	Exits interface configuration mode and enters global configuration mode.
Step 5	(Optional) switch(config)# **show policy-map** [**interface** *interface* | **vlan** *vlan_id*] [**input** | **output**] [**type qos** | **queuing**] [**class** [**type qos** | **queuing**] *class-map-name*]	Displays information about policy maps that are applied to all interfaces or the specified interface. You can limit what the device displays to input or output policies, qos or queuing polices, and to a specific class.

	Command or Action	Purpose
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Attaching a QoS Policy Action to a VLAN

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan configuration** *vlan-id-list*	Enters VLAN configuration mode. *vlan-id-list* is a space-separated list of VLANs.
Step 3	switch(config-vlan-config)#**service-policy** [**type qos**] {**input** | **output**} {*policy-map-name*} [**no-stats**]	Adds the policy map to the input or output packets of a VLAN. Only one input policy and one output policy can be attached to a VLAN.
Step 4	switch(config-if)# **exit**	Exits VLAN configuration mode and enters global configuration mode.
Step 5	(Optional) switch(config)# **show policy-map** [**interface** *interface* | **vlan** *vlan_id*] [**input** | **output**] [**type qos** | **queuing**] [**class** [**type qos** | **queuing**] *class-map-name*]	Displays information about policy maps that are applied to all interfaces or the specified interface. You can limit what the device displays to input or output policies, qos or queuing polices, and to a specific class.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the running configuration to the startup configuration.

Session Manager Support for QoS

Beginning in Cisco NX-OS Release 4.2, Session Manger supports the configuration of QoS. This feature allows you to verify the QoS configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For information about Session Manager, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

After you start the configuration session, do not enter any configuration commands using the configure terminal configuration mode until the configuration session is aborted or committed. Entering parallel configurations (one configuration that uses the configuration session and another using the configuration terminal configuration mode) might cause verification failures in the configuration session mode.

Feature History for Using Modular QoS CLI

Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release.

Table 6: Feature History for Modular QoS CLI

Feature Name	Release	Feature Information
No changes from Release 4.2(1)	5.1(1)	—
Support for Session Manager	4.2(1)	Allows you to verify the configuration and required resources prior to committing them to the running configuration.