Configuring Quality of Service (QoS)

QoS includes traffic classification, marking, policing, queuing, and scheduling configured with service policies that are attached to ingress and egress targets. With QoS, you can provide preferential treatment to certain types of traffic at the expense of other types. When you do not configure QoS, the switch offers best-effort service to each packet, regardless of the packet contents or size.

Ingress QoS includes classification, marking, and policing. Classification can be based on the class of service (CoS), Differentiated Services Code Point (DSCP) in the inbound packet. You can classify based on Layer 2 MAC, IP-standard, or match based on AMAC, IP parameters using QCE configurations.

For EVC level QoS, see Creating a Policer.

Hierarchical QoS on the Cisco ME 1200 Series Carrier Ethernet Access Devices supports queuing and scheduling per EVC level per port. 8 queues are supported per port on the EVC. The EVC must be configured on the interface before configuring HQoS.

Note Single EVC per single UNI (input port) is supported.

The port scheduler is configured by default for fair round-robin scheduling between each EVC and non-service traffic, but can be weighted by configuring guaranteed bandwidths for the EVC. When guaranteed bandwidth is configured for an EVC, the remaining bandwidth of the port is divided equally between the remaining EVCs and non-service traffic.

Note Configuring of guaranteed bandwidth for non-service traffic is *not* supported.

The CIR bandwidth requirements are configured for each CoS per EVC. Excess traffic is strictly *not* prioritized, but is shared proportionally between the CoS within the EVC.

For more information, see Configuring Quality of Service (QoS).

How to Configure QoS

Provisioning the Controller to Configure QoS

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**controller nid** *1/NID_ID* **Example:** `Switch(config)# controller nid 1/1`	Enters the controller configuration mode.
Step 3	**ProvisionQos** **Example:** `Switch(config-controller)# ProvisionQos`	Enters QoS provisioning mode.

Step 4	**ProvisionQos** {**default** | **deleteQCE** | **exit** | **getInputGlobalPolicy** | **getInputGlobalPolicy** | **getOutputGlobalPolicy** | **getOutputPortPolicy** | **getQCE** | **getSystemQosSettings** | **no** | **reorderQCEentries** | **setInputGlobalPolicy** | **setInputPortPolicy** | **setOutputGlobalPolicy** | **setOutputPortPolicy** | **setQCE** | **setSystemQosSettings** | **showQCElist** | **showQueueStatistics**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# ?`
	`ProvisionQos sub-mode commands:`
	`default Set a command to its defaults`
	`deleteQCE Delete a particular QCE`
	`exit Exit from ProvisionQos sub configuration mode`
	`getInputGlobalPolicy Show Output QoS global features configured`
	`getInputPortPolicy Show Input Policy configured on Physical Port`
	`getOutputGlobalPolicy Show Global Output QoS features`
	`getOutputPortPolicy Show Output Policy configured on Physical Port`
	`getQCE getQCE (default)`
	`getSystemQosSettings getSystemQosSettings (default)`
	`no Negate a command or set its defaults`
	`reorderQCEentries reorderQCEentries (default)`
	`setInputGlobalPolicy configure Global Input QoS features`
	`setInputPortPolicy configure Input policy on Physical Port`
	`setOutputGlobalPolicy configure Global Output QoS features`
	`setOutputPortPolicy configure Output policy on Physical Port`
	`setQCE setQCE (default)`
	`setSystemQosSettings set System-wide QoS settings`
	`showQCElist showQCElist (default)`
	`showQueueStatistics Display egress queue statistics`
		Displays the supported configurations for QoS.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The following example shows the supported QoS configuration:

```
Switch(config-controller-ProvisionQos)# ?
ProvisionQos sub-mode commands:
  default                Set a command to its defaults
  deleteQCE              Delete a particular QCE
  exit                   Exit from ProvisionQos sub configuration mode
  getInputGlobalPolicy   Show Output QoS global features configured
  getInputPortPolicy     Show Input Policy configured on Physical Port
  getOutputGlobalPolicy  Show Global Output QoS features
  getOutputPortPolicy    Show Output Policy configured on Physical Port
  getQCE                 getQCE (default)
  getSystemQosSettings   getSystemQosSettings (default)
  no                     Negate a command or set its defaults
  reorderQCEentries      reorderQCEentries (default)
  setInputGlobalPolicy   configure Global Input QoS features
  setInputPortPolicy     configure Input policy on Physical Port
  setOutputGlobalPolicy  configure Global Output QoS features
  setOutputPortPolicy    configure Output policy on Physical Port
  setQCE                 setQCE (default)
  setSystemQosSettings   set System-wide QoS settings
  showQCElist            showQCElist (default)
  showQueueStatistics    Display egress queue statistics
```

Configuring QoS Input Policy Features Globally on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setInputGlobalPolicy** {**commit** | **flush** | **inputGlobalPolicyConfig** | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# setInputGlobalPolicy ?`
	`  commit  commit setInputGlobalPolicy`
		Sets the global input QoS policy.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **inputGlobalPolicyConfig—**Sets the input policy global configuration on the controller.

	Command or Action	Purpose

	`flush flush all setInputGlobalPolicy commands from queue`
	`inputGlobalPolicyConfig configure Global Input QoS features`
	`review review setInputGlobalPolicy commands`
		• **review—**Displays the configuration on the controller.

Step 2	**setInputGlobalPolicy inputGlobalPolicyConfig** {**match_DSCP value_xx** {**mark_DSCP** *dscp_value* | **mark_egress_class** *egress_queue*} | **service_policy** {**attach** | **detach**}} **Example:** `Switch(config-controller-ProvisionQos)#` **`setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_00 mark_DSCP 4`** `Switch(config-controller-ProvisionQos)#` **`setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_00 mark_egress_class 4`** `Switch(config-controller-ProvisionQos)#` **`setInputGlobalPolicy inputGlobalPolicyConfig service_policy attach`**	Configures input global policy. • **match_DSCP—**Matches a particular DSCP value . • **value_xx—**Sets DSCP values on the controller. Use the following values: • **value_00** DSCP 0. Default PHB for best effort traffic **value_01** DSCP 1 **value_02** DSCP 2 **value_03** DSCP 3 **value_04** DSCP 4 **value_05** DSCP 5 **value_06** DSCP 6 **value_07** DSCP 7 **value_08** CS1. Class Selector PHB precedence 1(DSCP 8) **value_09** DSCP 9 **value_10** AF11. Assured Forwarding PHB (DSCP 10) **value_11** DSCP 11 **value_12** AF12. Assured Forwarding PHB (DSCP 12) **value_13** DSCP 13 **value_14** AF13. Assured Forwarding PHB (DSCP 14) **value_15** DSCP 15 **value_16** CS2. Class Selector PHB precedence 1(DSCP 16) **value_16** CS2. Class Selector PHB precedence 1(DSCP 16) **value_17** DSCP 17 **value_18** AF21. Assured Forwarding PHB (DSCP 18) **value_19** DSCP 19 **value_20** AF22. Assured Forwarding PHB (DSCP 20) **value_21** DSCP 21 **value_22** AF23. Assured Forwarding PHB (DSCP 22) **value_23** DSCP 23 **value_24** CS3. Class Selector PHB precedence 1(DSCP 24) **value_25** DSCP 25

	Command or Action	Purpose
		value_26 AF31. Assured Forwarding PHB (DSCP 26) **value_27** DSCP 27 **value_28** AF32. Assured Forwarding PHB (DSCP 28) **value_29** DSCP 29 **value_30** AF33. Assured Forwarding PHB (DSCP 30) **value_31** DSCP 31 **value_32** CS4. Class Selector PHB precedence 1(DSCP 32) **value_33**DSCP 33 **value_34** AF41. Assured Forwarding PHB (DSCP 34) **value_35** DSCP 35 **value_36** AF42. Assured Forwarding PHB (DSCP 36) **value_37** DSCP 37 **value_38** AF43. Assured Forwarding PHB (DSCP 38) **value_39** DSCP 39 **value_40** CS5. Class Selector PHB precedence 1(DSCP 40) **value_41** DSCP 41 **value_42** DSCP 42 **value_43** DSCP 43 **value_44** VA. Voice Admit PHB(DSCP 44) **value_45** DSCP 45 **value_46** Expedited Forwarding PHB(DSCP 46) **value_47** DSCP 47 **value_48** CS6. Class Selector PHB precedence 1(DSCP 48) **value_49** DSCP 49 **value_50** DSCP 50 **value_51** DSCP 51 **value_52** DSCP 52 **value_53** DSCP 53 **value_54** DSCP 54 **value_55** DSCP 55 **value_56** CS7. Class Selector PHB precedence 1(DSCP 56) **value_57** DSCP 57 **value_58** DSCP 58 **value_59** DSCP 59

	Command or Action	Purpose
		value_60 DSCP 60 **value_61** DSCP 61 **value_62** DSCP 62 **value_63** DSCP 63 • **mark_DSCP—**Marks the DSCP on the controller. The valid range is from 0 to 63. 64 is invalid. • **mark_egress_class—**Assigns to egress queue. The valid range is from 0 to 7. 8 is invalid. • **service_policy—**Attaches or detaches the service policy. ◦ **attach—**Attaches the service policy and enables the configuration. ◦ **detach—**Removes the service policy, removes all configuration and restore the default configuration.

Step 3	**setInputGlobalPolicy review**
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`setInputGlobalPolicy review`
	`Commands in queue:`
	`setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_02 mark_DSCP 4`
	`setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_02 mark_egress_class 4`
	`setInputGlobalPolicy inputGlobalPolicyConfig service_policy attach`
		Displays the QoS configuration on the controller.

Step 4	**setInputGlobalPolicy commit** **Example:** `Switch(config-controller-ProvisionQos)#` **`setInputGlobalPolicy commit`**	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)#` **`exit`**	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS input policy globally on the controller:

```
Switch(config-controller-ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig
match_DSCP value_00 mark_DSCP 4
Switch(config-controller-ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig
match_DSCP value_00 mark_egress_class 4
Switch(config-controller-ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig
service_policy attach
Switch(config-controller-ProvisionQos)# setInputGlobalPolicy review
Commands in queue:
        setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_02 mark_DSCP 4
        setInputGlobalPolicy inputGlobalPolicyConfig match_DSCP value_02 mark_egress_class
 4
        setInputGlobalPolicy inputGlobalPolicyConfig service_policy attach

Switch(config-controller-ProvisionQos)# setInputGlobalPolicy commit
SetInputGlobalPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring QoS Input Policy Features at Port level on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setInputPortPolicy** {**commit** | **flush** | **inputPortPolicyConfig** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setInputPortPolicy ?`
	`  commit  commit setInputPortPolicy commands from queue`
	`  flush  flush all setInputPortPolicy commands from queue`
	`  inputPortPolicyConfig  configure Input policy on Physical Port`
	`  review  review setInputPortPolicy commands`
		Sets the input QoS policy at port level.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **inputPortPolicyConfig—**Sets the input policy configuration at port level on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**inputPortPolicyConfig** {**egress_class_marking** {**enable** | **disable**} | **globalDscpBasedDscpIngressMarking** {**enable** | **disable**} | **globalDscpBasedEgressClassMarking** {**enable** | **disable**} | **match cos value_xx mark_egress_class** *egress_queue* | **port_number***port_number* | **port_policer** {**cir** *kbps* | **state** {**enable** | **disable**}} | **qce** {**address** {**destination** | **source**} | **key** {**double_tag** | **ip_address** | **mac_ip_addr** | **normal**}} | **service_policy** {**attach** | **detach**}}	Configures input policy at port. • **egress_class_marking—**Enables egress class marking as configured per each cos matched packet. • **globalDscpBasedDscpIngressMarking—**Enables DSCP based DSCP ingress marking on the port as per the configured global policy. • **globalDscpBasedEgressClassMarking—**Enable DSCP based Egress Class marking on the port as per the configured global policy.

Command or Action	Purpose
Example: `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig egress_class_marking enable`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig globalDscpBasedDscpIngressMarking enable`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig globalDscpBasedEgressClassMarking enable`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig match cos value_0`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig port_number 1`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig port_policer cir 1000`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig port_policer state enable`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig qce address destination`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig qce key double_tag`** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy inputPortPolicyConfig service_policy attach`**	• **enable—**Enables the marking. • **disable—**Disables the marking. • **match—**Matches input packet COS. • **cos—**Sets the COS value. ◦ **value_0—**Sets the COS value 0. ◦ **value_1—**Sets the COS value 1. ◦ **value_2—**Sets the COS value 2. ◦ **value_3—**Sets the COS value 3. ◦ **value_4—**Sets the COS value 4. ◦ **value_5—**Sets the COS value 5. ◦ **value_6—**Sets the COS value 6. ◦ **value_7—**Sets the COS value 7. • **mark_egress_class** *egress_queue***—**Sets the egress queue value. The valid range is from 0 to 7. Queue 8 in invalid. • **port_number** *port_number***—**Sets the port number. The valid range is from 1 to 6. Port 7 is invalid. • **port_policer—**Sets port level policer. • **cir** *kbps***—**Sets committed information rate in kbps. The valid range is from 100 to 1000000. • **state—**Enables or disables the policer state. • **qce—**Sets TCAM based QoS control entry settings for the port. • **address—**Matches the source or destination address of incoming packet in QCE. • **destination—**Matches against destination address. • **source—**Matches against source address. • **key—**Matches the key template in QCE. The default is normal. ◦ **double_tag—**Matches against Match outer tag, inner tag, IP protocol, DSCP and DPORT. ◦ **ip_address—**Matches against Match outer tag, SMAC/DMAC, IP protocol, DSCP, SIP and DIP.

	Command or Action	Purpose
		◦ **mac_ip_addr—**Match outer tag, inner tag, SMAC, DMAC, IP protocol, DSCP, SIP,DIP, SPORT and DPORT. ◦ **normal—**Match outer tag, SMAC/DMAC, IP protocol, DSCP, SIP/DIP, SPORT and DPORT (default). • **service_policy—**Attaches or detaches the service policy on the port. • **attach—**Attaches the service policy and enables the configuration. • **detach—**Removes the service policy, removes the configuration and restores the default configuration.

Step 3	**setInputPortPolicy review**
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`setInputPortPolicy review`
	`Commands in queue:`
	`Commands in queue:`
	`setInputPortPolicy inputPortPolicyConfig egress_class_marking enable`
	`setInputPortPolicy inputPortPolicyConfig globalDscpBasedDscpIngressMarking enable`
	`setInputPortPolicy inputPortPolicyConfig globalDscpBasedEgressClassMarking enable`
	`setInputPortPolicy inputPortPolicyConfig match cos value_0 mark_egress_class 5`
	`setInputPortPolicy inputPortPolicyConfig match cos value_0 mark_egress_class 5`
	`setInputPortPolicy inputPortPolicyConfig port_policer state enable`
	`setInputPortPolicy inputPortPolicyConfig port_number 1`
	`setInputPortPolicy inputPortPolicyConfig port_policer cir 1000`
	`setInputPortPolicy inputPortPolicyConfig port_policer state enable`
	`setInputPortPolicy inputPortPolicyConfig qce address source`
	`setInputPortPolicy inputPortPolicyConfig qce key normal`
	`setInputPortPolicy inputPortPolicyConfig service_policy attach`
		Displays the QoS configuration on the controller.

Step 4	**setInputPortPolicycommit** **Example:** `Switch(config-controller-ProvisionQos)#` **`setInputPortPolicy commit`**	Sends the QoS configuration to the NID.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS input port policy on the controller:

```
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
egress_class_marking enable
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
globalDscpBasedDscpIngressMarking enable
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
globalDscpBasedEgressClassMarking enable
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig match cos
 value_0
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port_number
 1
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port_policer
 cir 1000
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port_policer
 state enable
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce address
 destination
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce key
double_tag
Switch(config-controller-ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
service_policy attach
Switch(config-controller-ProvisionQos)# setInputPortPolicy review
Commands in queue:
        Commands in queue:
        setInputPortPolicy inputPortPolicyConfig egress_class_marking enable
        setInputPortPolicy inputPortPolicyConfig globalDscpBasedDscpIngressMarking enable
        setInputPortPolicy inputPortPolicyConfig globalDscpBasedEgressClassMarking enable
        setInputPortPolicy inputPortPolicyConfig match cos value_0 mark_egress_class 5
        setInputPortPolicy inputPortPolicyConfig match cos value_0 mark_egress_class 5
        setInputPortPolicy inputPortPolicyConfig port_policer state enable
        setInputPortPolicy inputPortPolicyConfig port_number 1
        setInputPortPolicy inputPortPolicyConfig port_policer cir 1000
        setInputPortPolicy inputPortPolicyConfig port_policer state enable
        setInputPortPolicy inputPortPolicyConfig qce address source
        setInputPortPolicy inputPortPolicyConfig qce key normal
        setInputPortPolicy inputPortPolicyConfig service_policy attach
Switch(config-controller-ProvisionQos)# setInputPortPolicy commit
SetInputPortPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring QoS Output Policy Features Globally on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setOutputGlobalPolicy** {**commit** | **flush** | **outputGlobalPolicyConfig** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy ?`
	`  commit  commit setOutputGlobalPolicy`
	`  flush  flush all setOutputGlobalPolicy commands from queue`
	`  outputGlobalPolicyConfig  configure Global Output QoS features`
	`  review  review setOutputGlobalPolicy commands`
		Sets the global output QoS policy.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **outputGlobalPolicyConfig—**Sets the output policy global configuration on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**setOutputGlobalPolicy outputGlobalPolicyConfig** {**match_DSCP value_xx** {**mark_DSCP** *dscp_value*} | **service_policy** {**attach** | **detach**}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig match_DSCP value_01 mark_DSCP 1`
	`Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig service_policy attach`
		Configures output global policy.
		• **match_DSCP—**Matches a particular DSCP value .
		• **value_xx—**Sets DSCP values on the controller. Use the following values:
		• **value_00** DSCP 0. Default PHB for best effort traffic
		value_01 DSCP 1
		value_02 DSCP 2
		value_03 DSCP 3
		value_04 DSCP 4
		value_05 DSCP 5
		value_06 DSCP 6
		value_07 DSCP 7
		value_08 CS1. Class Selector PHB precedence 1(DSCP 8)
		value_09 DSCP 9
		value_10 AF11. Assured Forwarding PHB (DSCP 10)
		value_11 DSCP 11
		value_12 AF12. Assured Forwarding PHB (DSCP 12)
		value_13 DSCP 13
		value_14 AF13. Assured Forwarding PHB (DSCP 14)
		value_15 DSCP 15
		value_16 CS2. Class Selector PHB precedence 1(DSCP 16)
		value_16 CS2. Class Selector PHB precedence 1(DSCP 16)
		value_17 DSCP 17

	Command or Action	Purpose
		value_18 AF21. Assured Forwarding PHB (DSCP 18) **value_19** DSCP 19 **value_20** AF22. Assured Forwarding PHB (DSCP 20) **value_21** DSCP 21 **value_22** AF23. Assured Forwarding PHB (DSCP 22) **value_23** DSCP 23 **value_24** CS3. Class Selector PHB precedence 1(DSCP 24) **value_25** DSCP 25 **value_26** AF31. Assured Forwarding PHB (DSCP 26) **value_27** DSCP 27 **value_28** AF32. Assured Forwarding PHB (DSCP 28) **value_29** DSCP 29 **value_30** AF33. Assured Forwarding PHB (DSCP 30) **value_31** DSCP 31 **value_32** CS4. Class Selector PHB precedence 1(DSCP 32) **value_33**DSCP 33 **value_34** AF41. Assured Forwarding PHB (DSCP 34) **value_35** DSCP 35 **value_36** AF42. Assured Forwarding PHB (DSCP 36) **value_37** DSCP 37 **value_38** AF43. Assured Forwarding PHB (DSCP 38) **value_39** DSCP 39 **value_40** CS5. Class Selector PHB precedence 1(DSCP 40) **value_41** DSCP 41 **value_42** DSCP 42 **value_43** DSCP 43 **value_44** VA. Voice Admit PHB(DSCP 44) **value_45** DSCP 45 **value_46** Expedited Forwarding PHB(DSCP 46) **value_47** DSCP 47 **value_48** CS6. Class Selector PHB precedence 1(DSCP 48) **value_49** DSCP 49 **value_50** DSCP 50 **value_51** DSCP 51

	Command or Action	Purpose
		value_52 DSCP 52 **value_53** DSCP 53 **value_54** DSCP 54 **value_55** DSCP 55 **value_56** CS7. Class Selector PHB precedence 1(DSCP 56) **value_57** DSCP 57 **value_58** DSCP 58 **value_59** DSCP 59 **value_60** DSCP 60 **value_61** DSCP 61 **value_62** DSCP 62 **value_63** DSCP 63 • **mark_DSCP—**Marks the DSCP on the controller. The valid range is from 0 to 63. 64 is invalid. • **service_policy—**Applies the service policy. • **attach—**Adds the service policy. • **detach—**Removes the service policy.

Step 3	**setOutputGlobalPolicy review**
	Example:
	`Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy review`
	`Commands in queue:`
	`setOutputGlobalPolicy outputGlobalPolicyConfig match_DSCP value_01 mark_DSCP 1`
	`setOutputGlobalPolicy outputGlobalPolicyConfig service_policy attach`
		Displays the QoS configuration on the controller.

Step 4	**setOutputGlobalPolicycommit** **Example:** `Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS output policy globally on the controller:

```
Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig
match_DSCP value_00 mark_DSCP 4
Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig
service_policy attach
Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy review
Commands in queue:
        setOutputGlobalPolicy outputGlobalPolicyConfig match_DSCP value_01 mark_DSCP 1
        setOutputGlobalPolicy outputGlobalPolicyConfig service_policy attach

Switch(config-controller-ProvisionQos)# setOutputGlobalPolicy commit
SetOutputGlobalPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring QoS Output Policy Features at Port level on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setOutputPortPolicy** {**commit** | **flush** | **outputPortPolicyConfig** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setOutputPortPolicy ?`
	`  commit  commit setOutputPortPolicy`
	`  flush  flush all setOutputPortPolicy commands from queue`
	`  outputPortPolicyConfig  configure Output policy on Physical Port`
	`  review  review setOutputPortPolicy commands`
		Sets the output QoS policy at port level.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **outputPortPolicyConfig—**Sets the output policy configuration at port level on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**outputPortPolicyConfig** {**cos_marking** {**enable** | **disable**} | **globalDscpBasedDscpEgressMarking** {**enable** | **disable**} | **matchegress_class_xx** {**bandwidth** {**priority_level** *pr_level* | **remaining_ratio** *ratio*} | **mark_cos** *mark_cos_vlaue* | **shaper** {**rate** *rate_value* | **state** {**enable** | **disable**}}} | **port_number***port_number* | **port_shape** {**rate** *rate_value* | **state** {**enable** | **disable**}} |**service_policy** {**attach** | **detach**}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig cos_marking enable`
	`Switch(config-controller-ProvisionQos)#`
		Configures output policy at port.
		• **cos_marking—**Enables egress class marking as configured per each cos matched packet.
		• **globalDscpBasedDscpEgressMarking—**Enables DSCP based DSCP egress marking on the port as per the configured global policy.
		• **enable—**Enables the marking.
		• **disable—**Disables the marking.
		• **match—**Matches output packet COS.

	Command or Action	Purpose
	setOutputPortPolicy outputPortPolicyConfig globalDscpBasedDscpEgressMarking enable Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig match egress_class_0 bandwidth priority_level 8** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig match egress_class_0 bandwidth remaining_ratio 20** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig match egress_class_0 mark_cos 7** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig port_number 1** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig port_shaper rate 1000** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig port_shaper state enable** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy outputPortPolicyConfig service_policy attach**	◦ **all-egress_classes—**Sets the egress for all queues from 0 to 7. ◦ **egress_class_0—**Sets queue 0, lowest priority. ◦ **egress_class_1—**Sets queue 1. ◦ **egress_class_2—**Sets queue 2. ◦ **egress_class_3—**Sets queue 3. ◦ **egress_class_4—**Sets queue 4. ◦ **egress_class_5—**Sets queue 5, higher priority. ◦ **egress_class_6—**Sets queue 6, highest priority. • **bandwith —**Sets scheduling scheme. • **priority_level** *pr_level***—**Configures priority scheduling. The valid range is from 1 to 8. • **remaining_ratio** *ratio***—**Configures weighted round robin mode of scheduling. The vallid range is from 1 to 100. • **shaper —**Configures queue level shaper. • **mark_cos** *cos_vallue***—**Sets the COS value for marking The valid range is from 0 to 7. COS 8 in invalid. • **port_number** *port_number***—**Sets the port number. The valid range is from 1 to 6. Port 7 is invalid. • **port_shaper—**Sets port level shaper. • **rate** *kbps***—**Sets shaper rate in kbps. The valid range is from 100 to 1000000. • **state—**Enables or disables the port shaper state. • **service_policy—**Sets the service policy on the port. • **attach—**Adds the service policy. • **detach—**Removes the service policy.
Step 3	**setOutputPortPolicy review** **Example:** Switch(config-controller-ProvisionQos)# **setOutputPortPolicy review** Commands in queue: setOutputPortPolicy outputPortPolicyConfig cos_marking enable	Displays the QoS configuration on the controller.

	Command or Action	Purpose
	setOutputPortPolicy outputPortPolicyConfig globalDscpBasedDscpEgressMarking enable setOutputPortPolicy outputPortPolicyConfig match egress_class_0 bandwidth priority_level 8 setOutputPortPolicy outputPortPolicyConfig match all-egress_classes mark _cos 7 setOutputPortPolicy outputPortPolicyConfig port_number 4 setOutputPortPolicy outputPortPolicyConfig port_shaper rate 100 setOutputPortPolicy outputPortPolicyConfig port_shaper state enable setOutputPortPolicy outputPortPolicyConfig service_policy attach	
Step 4	**setOutputPortPolicycommit** **Example:** `Switch(config-controller-ProvisionQos)# setOutputPortPolicy commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS output port policy on the controller:

```
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig cos_marking
 enable
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig
globalDscpBasedDscpEgressMarking enable
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match
egress_class_0 bandwidth priority_level 8
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match
egress_class_0 bandwidth remaining_ratio 20
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match
egress_class_0 mark_cos 7
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port_number
 1
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port_shaper
 rate 1000
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port_shaper
 state enable
Switch(config-controller-ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig
service_policy attach
Switch(config-controller-ProvisionQos)# setOutputPortPolicy review
Commands in queue:
        setOutputPortPolicy outputPortPolicyConfig cos_marking enable
        setOutputPortPolicy outputPortPolicyConfig globalDscpBasedDscpEgressMarking enable
        setOutputPortPolicy outputPortPolicyConfig match egress_class_0 bandwidth
priority_level 8
        setOutputPortPolicy outputPortPolicyConfig match all-egress_classes mark _cos 7
        setOutputPortPolicy outputPortPolicyConfig port_number 4
        setOutputPortPolicy outputPortPolicyConfig port_shaper rate 100
        setOutputPortPolicy outputPortPolicyConfig port_shaper state enable
        setOutputPortPolicy outputPortPolicyConfig service_policy attach
Switch(config-controller-ProvisionQos)# setOutputPortPolicy commit
```

```
SetInputPortPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring QoS Control Entry (QCE) on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE** {**commit** | **flush** | **QCE_configuration** | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# setQCE ?`
	`  QCE_configuration  setQCE (default)`
	`  commit  commit setQCE`
	`  flush  flush all setQCE commands from queue`
	`  review  review setQCE commands`
		Sets QCE configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **QCE_configuration—**Sets the default QCE on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS QCE on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4
Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type
any match_fields inner_cos val_0-1
Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type
any match_type vlan c_tagged
Switch(config-controller-ProvisionQos)# setQCE review

Commands in queue:
        setQCE QCE_configuration control actions mark_COS 4
        setQCE QCE_configuration control actions mark_DSCP 3
        setQCE QCE_configuration control actions mark_egress_class 4
        setQCE QCE_configuration control ingress_match frame_type any
       setQCE QCE_configuration control ingress_match inner_tag_match match_fields inner_cos
 val_0-1

Switch(config-controller-ProvisionQos)# setQCE commit
 SetQCE Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring QoS Control Entry (QCE) Control Actions on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
		Configures QCE.
		
		• **control—**Configures QCE.
		
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.

	Command or Action	Purpose
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`	• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configurationcontrol** {**actions** {**mark_COS** *cos_vlaue* | **mark_DSCP** *dscp_vlaue* | **mark_egress_class** *egress_queue*}
	
	Example:
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_DSCP 3`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_egress_class 4`
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions.
		• **mark_COS** *cos_vlaue***—**Marks the Cos packets. The valid range is from 0 to 7. Value 8 is invalid.
		• **mark_DSCP** *dscp_vlaue***—**Marks the DSCP packets. The valid range is from 0 to 63. Value 64 is invalid.
		• **mark_egress_class** *egress_queue***—**Marks the egress queue. The valid range is from 0 to 7. Value 8 is invalid.

Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QoS input policy globally on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCEConfig match_DSCP value_00 mark_DSCP 4
Switch(config-controller-ProvisionQos)# setQCE QCEConfig match_DSCP value_00 mark_egress_class
 4
Switch(config-controller-ProvisionQos)# setQCE QCEConfig service_policy attach
Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
        setQCE QCE_configuration control actions mark_COS 4
        setQCE QCE_configuration control actions mark_DSCP 3
        setQCE QCE_configuration control actions mark_egress_class 4

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Match Ingress Parameters on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration {control {actions | ingress_match} | qce-id}**
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		
		• **control—**Configures QCE.
		
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configuration control ingress_match {frame_type | inner_tag_match | mac_params outer_tag_match | ports}**
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match frame_type any match_fields inner_cos val_0-1 any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match inner_tag_match match_fields inner_cos val_0-3`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match mac_params dmac_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match outer_tag_match match_fields cos val_2-3`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match ports gigabitEthernet_2 enable`
		• **ingress_match—**Configures ingress match.
		
		◦ **frame_type—**Matches against frame payload. See Configuring QCE Control Ingress Match Frame Type Parameter on the Controller, on page 21.
		
		◦ **inner_tag_match—**Matches against inner tag. See Configuring QCE Control Ingress Inner Tag Match Parameter on the Controller, on page 25.
		
		◦ **mac_params—**Matches against MAC filters. See Configuring QCE Control Ingress MAC Params Parameter on the Controller, on page 27.
		
		◦ **outer_tag_match—**Matches against outer tag. See Configuring QCE Control Ingress Outer Tag Match Parameter on the Controller, on page 29.
		
		◦ **ports—**Matches against ports. See Configuring QCE Control Ingress Ports Parameter on the Controller, on page 31.

	Command or Action	Purpose
Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE control ingress match parameters on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE-configuration control ingress_match
frame_type any match_fields inner_cos val_0-1 any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match
inner_tag_match match_fields inner_cos val_0-3
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match
mac_params dmac_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match
outer_tag_match match_fields cos val_2-3
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration control ingress_match ports
 gigabitEthernet_2 enable
Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
       setQCE QCE_configuration control ingress_match inner_tag_match match_fields inner_cos
 val_0-3
        setQCE QCE_configuration control ingress_match mac_params dmac_filter any
        setQCE QCE_configuration control ingress_match outer_tag_match match_fields cos
val_2-3
        setQCE QCE_configuration control ingress_match ports GigabitEthernet_2 enable

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Control Ingress Match Frame Type Parameter on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configuration control ingress_match frame_type** {**any** | **ipv4** {**dest_ip_filter** {**any** | **host** *host_name* | **network** {**dest_ip_addr** *dest_add* | **dest_ip_mask** *dest_mask*}} | **dscp_filter** {**any** | **range** *range_value* | **specific** *dscp_filter*} | **fragment_type** {**any** | **frag** | **non_frag**} | **protocol** {**any** | **specific** *protocol_value* | **tcp** | **udp**} | **source_ip_filter** {**any** | **host** *host_name* | **network** {**source_ip_addr source_ip_addr** *source_add* | **source_ip_mask** *source_mask*}} | **ipv6** {**dest_ip_filter** {**any** | **specific** {**dest_ip_addr_32bits** *dest_add* | **dest_ip_mask_32bits** *dest_mask*}} | **dscp_filter** {**any** | **range** *range_value* | **specific** *dscp_filter*} | **protocol** {**any** | **specific** *protocol_value* | **tcp** | **udp**} | **source_ip-filter** {**any** | **specific source_ip_addr_32bits** *source_add* | **source_ip_mask_32bits** *source_mask*}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 dest_ip_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 dest_ip_filter host host1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 dest_ip_filter network dest_ip_addr addr2`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 dscp_filter host any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 dscp_filter range 3-4`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 fragment_type frag`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type ipv4 protocol specific 45`
		• **control—**Configures QCE .
		• **ingress_match—**Configures ingress match.
		◦ **frame_type—**Matches against frame payload.
		◦ **any—**Matches against any frame payload .
		◦ **ipv4—**Matches against IPv4 frames.
		• **dest_ip_filter—**Matches against destination IP address filter .
		• **dscp_filter—**Matches against DSCP filter .
		• **fragment_type—**Matches against fragment type filter .
		• **protocol—**Matches against protocol filter .
		• **source_ip_filter—**Matches against source IP address filter .
		◦ **ipv6—**Matches against IPv6 frames .
		◦ **any—**Matches against any IP address, or filter.
		◦ **host** *host_name*—Matches against a specified host .
		◦ **network—**Matches against a network .
		◦ **dest_ip_addr** *dest_add*—Matches against the destination IP address .

	Command or Action	Purpose
	Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv4 source_ip_filter network source_ip_mask soumask** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 dest_ip_filter any** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 dest_ip_filter specfic dest_ip_addr_32its dest34** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 dscp_filter specific 45 any** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 protocol specific 450** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 protocol specific 45** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match frame_type ipv6 source_ip_filter specific source_ip_mask source-mask**	◦ **dest_ip_mask** *dest_mask*—Matches against the destination IP address mask. ◦ **range** *range_value*—Matches against the specified range . ◦ **specific** *dscp_filter*—Matches against the specific DSCP filter . ◦ **frag**—Matches against the specified IP fragment type . ◦ **non_frag**—Matches against the non fragment type . ◦ **specific** *protocol_value*—Matches against the specific protocol value . ◦ **tcp**—Matches against the TCP value . ◦ **udp**—Matches against the UDP value . ◦ **source_ip_addr** *source_addr*—Matches against the source IP address . ◦ **source_ip_mask** *source_mask*—Matches against the source IP address mask. ◦ **dest_ip_addr_32bits** *dest_add*—Matches against the destination IP address. ◦ **dest_ip_mask_32bits** *dest_mask*—Matches against the destination IP address mask. ◦ **source_ip_addr_32bits** *source_add*—Matches against the source IP address. ◦ **source_ip_mask_32bits** *source_mask*—Matches against the source IP address mask.
Step 3	**setQCE review** **Example:** Switch(config-controller-ProvisionQos)# **setQCE review**	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** Switch(config-controller-ProvisionQos)# **setQCE commit**	Sends the QoS configuration to the NID.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE Control Ingress Match frame type parameters on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 dest_ip_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 dest_ip_filter host host1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 dest_ip_filter network dest_ip_addr addr2
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 dscp_filter host any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 dscp_filter range 3-4
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 fragment_type frag
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 protocol specific 45
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv4 source_ip_filter network source_ip_mask soumask
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 dest_ip_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 dest_ip_filter specfic dest_ip_addr_32its dest34
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 dscp_filter specific 45 any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 protocol specific 450
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 protocol specific 45
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match frame_type
ipv6 source_ip_filter specific source_ip_mask source-mask

Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
       setQCE QCE_configuration control ingress_match inner_tag_match match_fields inner_cos
 val_0-3
        setQCE QCE_configuration control ingress_match outer_tag_match match_fields cos
val_2-3
        setQCE QCE_configuration control ingress_match frame_type ipv6 dest_ip_filter any
       setQCE QCE_configuration control ingress_match frame_type ipv6 dscp_filter specific
 45
       setQCE QCE_configuration control ingress_match frame_type ipv6 protocol pecific 450

        setQCE QCE_configuration control ingress_match frame_type ipv6 source_ip_filter
specific source_ip_mask_32bits source-mask

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Control Ingress Inner Tag Match Parameter on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configuration control ingress_match inner_tag_match** {**match_fields** {**inner_cos inner_cos_xx** | **vlan_id_filter** {**any** | **range** *vlan_range* | **specifc** *specific_vlan*}} | **match_type** {**any** | **c-tagged** | **s_tagged** | **tagged** | **untagged**}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match match_fields vlan_id_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match match_fields vlan_id_filter range range1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match match_fields vlan_id_filter specific 450`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match match_type c_tagged`
		• **control—**Configures QCE.
		• **ingress_match—**Configures ingress match.
		◦ **inner_tag_match—**Matches against inner tag value.
		◦ **match_fields—**Matches against tag fields .
		◦ **inner-cos inner_cos_xx—**Matches against inner packet Cos value.
		◦ **val_0-1—**Specifies packet COS 0-1.
		◦ **val_0-3—**Specifies packet COS 0-3.
		◦ **val_0-only—**Specifies packet COS 0.
		◦ **val_1-only—**Specifies packet COS 1.
		◦ **val_2-3—**Specifies packet COS 2-3.
		◦ **val_2-only—**Specifies packet COS 2-only.
		◦ **val_3-only—**Specifies packet COS 3-only.
		◦ **val_4-5—**Specifies packet COS 4-5.
		◦ **val_4-7—**Specifies packet COS 4-7.
		◦ **val_4-only—**Specifies packet COS 4-only.

	Command or Action	Purpose
		◦ **val_5-only—**Specifies packet COS 5-only. ◦ **val_6-7—**Specifies packet COS 6-7. ◦ **val_6-only—**Specifies packet COS 6. ◦ **val_7-only—**Specifies packet COS 7. ◦ **val_any—**Specifies packet COS any. ◦ **vlan_id_filter—**Matches against VLAN ID filter. ◦ **any—**Matches against any VLAN. ◦ **range** *vlan_range***—**Matches against the specified VLAN range . ◦ **specific** *specific_vlan***—**Matches against the specific VLAN. The valid range is from 1 to 4095. ◦ **match_type—**Matches against tag fields. • **any—**Matches against any tagged . • **c-tagged—**Matches against C tagged . • **s-tagged—**Matches against S tagged . • **tagged—**Matches against tagged . • **untagged—**Matches against untagged .
Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE Control Match Ingress inner tag parameters on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields inner_cos val_0-1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter range range1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter specific 450
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_type c_tagged

Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
       setQCE QCE_configuration control ingress_match inner_tag_match match_fields inner_cos
 val_0-1
        setQCE QCE_configuration control ingress_match inner_tag_match match_fields
vlan_id_filter any
        setQCE QCE_configuration control ingress_match inner_tag_match match_fields
vlan_id_filter range range1
        setQCE QCE_configuration control ingress_match inner_tag_match match_fields
vlan_id_filter specific 450
        setQCE QCE_configuration control ingress_match inner_tag_match match_type c_tagged

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Control Ingress MAC Params Parameter on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		• **control**—Configures QCE.
		• **actions**—Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		• **ingress_match**—Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		• **qce-id**—Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configuration** {**control ingress_match mac_params** {**dmac_filter** {**any** | **broadcast** | **multicast** | **specific**	• **control**—Configures QCE. • **ingress_match**—Configures ingress match.

	Command or Action	Purpose

	specific_fillter | **unicast**} |**smac_filter** {**any** | **specific** *specific_filter*}
	
	Example:
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match mac_params dmac_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match mac_params dmac_filter broadcast`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match mac_params dmac_filter specific filter1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match mac_params smac_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match mac_params smac_filter specific filter2`
		• **mac_params—**Configures MAC filters.
		• **dmac_filter—**Configures destination MAC filters.
		• **smac_filter—**Configures source MAC filters.
		• **any—**Configures any MAC filter.
		• **broadcast—**Configures any broadcast MAC filter.
		• **multicast—**Configures any multicast MAC filter.
		• **specific** *specific_filter***—**Configures specific MAC filter.

Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE Control Match Ingress MAC params parameters on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields inner_cos val-0-1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter range range1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_fields vlan_id_filter specific 450
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match inner_tag_match
 match_type c-tagged

Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
        setQCE QCE_configuration control ingress_match mac_params dmac_filter broadcast
        setQCE QCE_configuration control ingress_match mac_params smac_filter specific
source1
        setQCE QCE_configuration control ingress_match mac_params dmac_filter specific
```

```
filter1
        setQCE QCE_configuration control ingress_match mac_params smac_filter specific
filter2

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Control Ingress Outer Tag Match Parameter on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged`
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		Configures QCE.
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20.
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 2	**setQCE QCE_configuration** {**control ingress_match outer_tag_match** {**match_fields** {**cos cos_xx** | **vlan_id_filter** {**any** | **range** *vlan_range* | **specifc** *specific_vlan*} | **match_type** {**any** | **c-tagged** | **s_tagged** | **tagged** | **untagged**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match match_fields cos val_0-1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match match_fields vlan_id_filter any`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match match_fields vlan_id_filter range range1`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match match_fields vlan_id_filter specific 230`
	`Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match match_type c-tagged`
		• **control—**Configures QCE.
		• **ingress_match—**Configures ingress match.
		◦ **outer_tag_match—**Matches against the outer tag value.
		◦ **match_fields—**Matches against outer tag fields .
		◦ **cos cos_xx—**Matches against packet Cos value.
		◦ **val_0-1—**Specifies packet COS 0-1.
		◦ **val_0-3—**Specifies packet COS 0-3.
		◦ **val_0-only—**Specifies packet COS 0.
		◦ **val_1-only—**Specifies packet COS 1.
		◦ **val_2-3—**Specifies packet COS 2-3.
		◦ **val_2-only—**Specifies packet COS 2-only.
		◦ **val_3-only—**Specifies packet COS 3-only.

	Command or Action	Purpose
		◦ **val_4-5—**Specifies packet COS 4-5. ◦ **val_4-7—**Specifies packet COS 4-7. ◦ **val_4-only—**Specifies packet COS 4-only. ◦ **val_5-only—**Specifies packet COS 5-only. ◦ **val_6-7—**Specifies packet COS 6-7. ◦ **val_6-only—**Specifies packet COS 6. ◦ **val_7-only—**Specifies packet COS 7. ◦ **val_any—**Specifies packet COS any. ◦ **vlan_id_filter—**Matches against VLAN ID filter. ◦ **any—**Matches against any VLAN. ◦ **range** *vlan_range***—**Matches against the specified VLAN range . ◦ **specific** *specific_vlan***—**Matches against the specific VLAN. The valid range is from 1 to 4095. ◦ **match_type—**Matches against tag fields. • **any—**Matches against any tagged . • **c-tagged—**Matches against C tagged . • **s-tagged—**Matches against S tagged . • **tagged—**Matches against tagged . • **untagged—**Matches against untagged .
Step 3	**setQCE review** **Example:** `Switch(config-controller-ProvisionQos)# setQCE review`	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** `Switch(config-controller-ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE Control Match Ingress outer tag parameters on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match
 match_fields cos val_0-1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match
 match_fields vlan_id_filter any
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match
 match_fields vlan_id_filter range range1
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match
 match_fields vlan_id_filter specific 230
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match outer_tag_match
 match_type c-tagged

Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
       setQCE QCE_configuration control ingress_match outer_tag_match match_fields
vlan_id_filter specific 230
        setQCE QCE_configuration control ingress_match outer_tag_match match_fields
vlan_id_filter range vlan2
       setQCE QCE_configuration control ingress_match outer_tag_match match_fields cos
val_0-1
        setQCE QCE_configuration control ingress_match outer_tag_match match_type c_tagged
Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring QCE Control Ingress Ports Parameter on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE QCE_configuration** {**control** {**actions** | **ingress_match**} | **qce-id**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setQCE QCEconfiguration control action mark_Cos 4`
	`Switch(config-controller-ProvisionQos)# setQCE`
		Configures QCE.
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the Controller, on page 18.

	Command or Action	Purpose
	QCEconfiguration ingress_match frame_type any match_fields inner_cos val_0-1 Switch(config-controller-ProvisionQos)# **setQCE QCEconfiguration ingress_match frame_type any match_type vlan c_tagged** Switch(config-controller-ProvisionQos)# **setQCE QCEconfiguration qce-id 4**	• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the Controller, on page 20. • **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.
Step 2	**setQCE QCE_configuration control ingress_match ports {GigabitEthernet_1 | GigabitEthernet_2 | GigabitEthernet_3 | GigabitEthernet_4 | GigabitEthernet_5 | GigabitEthernet_6} {enable | disable}** **Example:** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match ports GigabitEthernet_1 enable** Switch(config-controller-ProvisionQos)# **setQCE QCE_configuration ingress_match ports GigabitEthernet_3 disable**	• **control—**Configures QCE. • **ingress_match—**Configures ingress match. • **ports—**Configures ingress ports. • **GigabitEthernet_1—**Configures physical port 1. • **GigabitEthernet_2—**Configures physical port 2. • **GigabitEthernet_3—**Configures physical port 3. • **GigabitEthernet_4—**Configures physical port 4. • **GigabitEthernet_5—**Configures physical port 5. • **GigabitEthernet_6—**Configures physical port 6. • **enable—**Enables the port. • **disable—**Disables the port.
Step 3	**setQCE review** **Example:** Switch(config-controller-ProvisionQos)# **setQCE review**	Displays the QoS configuration on the controller.
Step 4	**setQCE commit** **Example:** Switch(config-controller-ProvisionQos)# **setQCE commit**	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** Switch(config-controller-ProvisionQos)# **exit**	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QCE Control Match Ingress ports on the controller:

```
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match ports
GigabitEthernet_1 enable
Switch(config-controller-ProvisionQos)# setQCE QCE_configuration ingress_match ports
GigabitEthernet_3 disable
```

```
Switch(config-controller-ProvisionQos)# setQCE review
Commands in queue:
        setQCE QCE_configuration control ingress_match ports GigabitEthernet_1 enable
        setQCE QCE_configuration control ingress_match ports GigabitEthernet_3 disable

Switch(config-controller-ProvisionQos)# setQCE commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring System QoS on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setSystemQosSettings** {**commit** | **flush** | **system_qos_config** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)#`
	`setSystemQosSettings ?`
	`  commit  commit setSystemQosSettings`
	`  flush  flush all setSystemQosSettings commands from queue`
	`  review  review setSystemQosSettings commands`
	`  system_qos_config  set System-wide QoS settings`
		Sets the system Qos configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **system_qos_config—**Sets the system wide QoS settings on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**setSystemQosSettings system_qos_config WRED** {**egress_class_0** | **egress_class_1** | **egress_class_2** | **egress_class_3** | **egress_class_4** | **egress_class_5**} {**max_threshold** *threshold_value*| **min_threshold** *threshold_value* | **state** {**enable** | **disable**}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)#`
	`setSystemQosSettings system_qos_config WRED egress_class_0 max_threshold 20`
	`Switch(config-controller-ProvisionQos)#`
	`setSystemQosSettings system_qos_config WRED egress_class_1 min_threshold 40`
	`Switch(config-controller-ProvisionQos)#`
	`setSystemQosSettings system_qos_config WRED egress_class_2 state enable`
		Configures system QoS.
		• **WRED—**Enables WRED algorithm for a non-priority queues on all ports.
		• **egress_class_0—**Egress queue 0.
		• **egress_class_1—**Egress queue 1.
		• **egress_class_2—**Egress queue 2.
		• **egress_class_3—**Egress queue 3.
		• **egress_class_4—**Egress queue 4.
		• **egress_class_5—**Egress queue 5.
		• **max_threshold** *threshold_value***—**Sets the maximum threshold.
		• **min_threshold** *threshold_value***—**Sets the minimum threshold.
		• **state—**Sets the WRED state per queue.

	Command or Action	Purpose
		• **enable—**Enables the WRED. • **disable—**Disables the WRED.

Step 3	**setSystemQosSettings review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`setSystemQosSettings review`
	
	`Commands in queue:`
	`setSystemQosSettings system_qos_config WRED egress_class_0 max_threshold 20`
	`setSystemQosSettings system_qos_config WRED egress_class_1 min_threshold 40`
	`setSystemQosSettings system_qos_config WRED egress_class_2 state enable`
		Displays the QoS configuration on the controller.

Step 4	**setSystemQosSettingscommit** **Example:** `Switch(config-controller-ProvisionQos)#` **`setSystemQosSettings commit`**	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)#` **`exit`**	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure QOS system settings on the controller:

```
Switch(config-controller-ProvisionQos)# setSystemQosSettings system_qos_config WRED
egress_class_0 max_threshold 20
Switch(config-controller-ProvisionQos)# setSystemQosSettings system_qos_config WRED
egress_class_1 min_threshold 40
Switch(config-controller-ProvisionQos)# setSystemQosSettings system_qos_config WRED
egress_class_2 state enable
Switch(config-controller-ProvisionQos)# setSystemQosSettings review

Commands in queue:
        setSystemQosSettings system_qos_config WRED egress_class_0 max_threshold  20
        setSystemQosSettings system_qos_config WRED egress_class_1 min_threshold  40
        setSystemQosSettings system_qos_config WRED egress_class_2 state enable
Switch(config-controller-ProvisionQos)# setSystemQosSettings commit
Switch(config-controller-ProvisionQos)# exit
```

Configuring Hierarchical QoS on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setsetHqosId** {**commit** | **flush** | **hqos_id_config** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setHqosId ?`
	`  commit  commit setHqosId`
	`  flush  flush all setHqosId commands from queue`
	`  hqos_id_config  setHqosId (default)`
	`  review  review setHqosId commands`
		Sets the hierarchical QoS configuration.
		
		• **commit—**Sends the QoS configuration to NID.
		
		• **flush—**Flushes all QoS configuration from the queue.
		
		• **hqos_id_config—**Sets the hierarchical QoS ID on the controller.
		
		• **review—**Displays the configuration on the controller.

Step 2	**setsetHqosId hqos_id_config** {**bandwidth** {**rate** *kbps* | **state** {**enable** | **disable**}} | **hqos_id** *hqos-id* | **match** {**egress_class_0** | **egress_class_1** | **egress_class_2** | **egress_class_3** | **egress_class_4** | **egress_class_5** | **egress_class_6** | **egress_class_7**} {**bandwidth priority-level** *priority*} | **shaper** {**rate** *kbps* | **state** {**enable** | **disable**}} | **port_number** *port-number* | **shaper** {**rate** *kbps* | **state** {**enable** | **disable**}}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config bandwidth rate 100`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config bandwidth state enable`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config hqos_id 4`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7 bandwidth priority_level 1`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7 shaper rate 100`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7 shaper state enable`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config port 2`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config shaper rate 100`
	`Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config shaper state enable`
		Configures hierarchical QoS.
		
		• **bandwidth—**Specifies bandwidth for logical interface.
		
		• **rate—**Specifies bandwidth rate in kbps. The valid range if from 100 to 1000000.
		
		• **state—**Specifies bandwidth state.
		
		• **enable—**Enables bandwidth state.
		
		• **disble—**Disables bandwidth state.
		
		• **hqos_id** *hqos-id*—Specifies HQos ID. The valid range is 0 to 256. 0 is invalid.
		
		• **match** —Specifies HQoS match queues.
		
		• **egress_class_0—**Egress queue 0; lowest priority
		
		• **egress_class_1—**Egress queue 1.
		
		• **egress_class_2—**Egress queue 2.
		
		• **egress_class_3—**Egress queue 3.
		
		• **egress_class_4—**Egress queue 4.
		
		• **egress_class_5—**Egress queue 5.
		
		• **egress_class_6—**Egress queue 6; higher priority.

	Command or Action	Purpose
		• **egress_class_7—**Egress queue 7; highest priority. • **bandwidhtpriority-level***priority***—**Sets the bandwidth priority scheduling level in strict mode. The valid values are 1-1. • **shaper—**Sets the queue level shaper. • **port_number** *port-number***—**Sets the port number. The valid range is from 1 to 6. Port 7 is invlaid. • **shaper—**Sets the interface level shaper. • **disable—**Disables the WRED.

Step 3	**setHqosId review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# setHqosId review`
	`Commands in queue:`
	`setHqosId hqos_id_config bandwidth rate 100`
	`setHqosId hqos_id_config bandwidth state enable`
	
	`setHqosId hqos_id_config hqos_id 4`
	`setHqosId hqos_id_config match egress_class_7 bandwidth priority_level 1`
	
	`setHqosId hqos_id_config match egress_class_7 shaper rate 100`
	`setHqosId hqos_id_config match egress_class_7 shaper state enable`
	`setHqosId hqos_id_config port_number 2`
	`setHqosId hqos_id_config shaper rate 100`
		Displays the HQoS configuration on the controller.

Step 4	**setHqosIdcommit** **Example:** `Switch(config-controller-ProvisionQos)# setHqosId commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure HQOS on the controller:

```
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config bandwidth rate 100
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config bandwidth state enable
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config hqos_id 4
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7
bandwidth priority_level 1
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7 shaper
 rate 100
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config match egress_class_7 shaper
```

```
 state enable
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config port 2
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config shaper rate 100
Switch(config-controller-ProvisionQos)# sethqosid hqos_id_config shaper state enable
Switch(config-controller-ProvisionQos)# setHqosId review
Commands in queue:
        setHqosId hqos_id_config bandwidth rate 100
        setHqosId hqos_id_config bandwidth state enable
        setHqosId hqos_id_config hqos_id 4
        setHqosId hqos_id_config match egress_class_7 bandwidth priority_level 1

        setHqosId hqos_id_config match egress_class_7 shaper rate 100
        setHqosId hqos_id_config match egress_class_7 shaper state enable
        setHqosId hqos_id_config port_number 2
        setHqosId hqos_id_config shaper rate 100
Switch(config-controller-ProvisionQos)# setHqosId commit
SetHqosId Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Configuring EVC Hierarchical QoS Policy on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setEvcHqosPolicy** {**commit** | **flush** | **hqos_id_config** | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setEvcHqosPolicy ?`
	`  commit  commit setEvcHqosPolicy`
	`  evcHqosPolicyConfig  setEvcHqosPolicy (default)`
	`  flush  flush all setEvcHqosPolicy commands from queue`
	`  review  review setEvcHqosPolicy commands`
		Sets the hierarchical QoS configuration on the EVC.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **evcHqosPolicyConfig—**Sets the EVC HQoS policy on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**setEvcHqosPolicy evcHqosPolicyConfig** {**evc_id** *evc-id* | **hqos_id** *hqos-id* | **service_policy** {**attach** | **detach**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig evc_id 1`
	`Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig hqos_id 2`
	`Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig service_policy attach`
		Configures hierarchical QoS on the EVC.
		• **evc_id** *evc-id—*Specifies EVC ID. The valid range is from 1 to 1024
		• **hqos_id** *hqos-id—*Specifies HQos ID. The valid range is 0 to 256. 0 is invalid.
		• **service_policy** —Specifies service policy that should be applied or removed on the EVC.
		• **attach—**Applies the policy on the EVC.
		• **detach—**Detaches the policy on the EVC.

	Command or Action	Purpose

Step 3	**setEvcHqosPolicy review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# setEvcHqosPolicy review`
	`Commands in queue:`
	`setEvcHqosPolicy evcHqosPolicyConfig evc_id 1`
	`setEvcHqosPolicy evcHqosPolicyConfig service_policy attach`
	`setEvcHqosPolicy evcHqosPolicyConfig service_policy detach`
	`setEvcHqosPolicy evcHqosPolicyConfig hqos_id 2`
	`setEvcHqosPolicy evcHqosPolicyConfig evc_id 1`
		Displays the HQoS EVC configuration on the controller.

Step 4	**setEvcHqosPolicycommit** **Example:** `Switch(config-controller-ProvisionQos)# setEvcHqosPolicy commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to configure EVC HQOS on the controller:

```
Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig evc_id 1
Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig hqos_id 2
Switch(config-controller-ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig attach
Switch(config-controller-ProvisionQos)# setEvcHqosPolicy review
Commands in queue:
        setEvcHqosPolicy evcHqosPolicyConfig evc_id 1
        setEvcHqosPolicy evcHqosPolicyConfig service_policy attach
        setEvcHqosPolicy evcHqosPolicyConfig service_policy detach
        setEvcHqosPolicy evcHqosPolicyConfig hqos_id 2
        setEvcHqosPolicy evcHqosPolicyConfig evc_id 1
Switch(config-controller-ProvisionQos)# setEvcHqosPolicy commit
Switch(config-controller-ProvisionQos)# exit
```

Reordering QoS Control Entry (QCE) on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**reorderQCEentries** {**commit** | **flush** | **reorder_qce** | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`reorderQCEentries ?`
	`  commit  commit reorderQCEentries`
	`  flush  flush all reorderQCEentries commands from queue`
	`  reorder_qce  reorderQCEentries (default)`
	`  review  review reorderQCEentries commands`
		Reorders the QCE entries.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **reorder_qce—**Reorders the QCE on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**reorderQCEentries reorder_qce** {**qce_id** *qce_id* | **reorder** {**after** | **before** | **last**} {**qce_id** *qce_id*}}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`reorderQCEentries reorder_qce qce_id 3`
	`Switch(config-controller-ProvisionQos)#`
	`reorderQCEentries reorder reorder before qce_id 2`
		• **reorder_qce—**Reorders QCE .
		• **reorder—**Specifies the reorder operation.
		• **after—**Reorders after the specified QCE ID.
		• **before—**Reorders before the specified QCE ID..
		• **last—**Reorders QCE ID as last.
		• **qce-id—** Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

Step 3	**reorderQCEentries review review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`reorderQCEentries review`
	`Commands in queue:`
	`  reorderQCEentries reorder_qce qce_id 3`
	`  reorderQCEentries reorder_qce qce_id 3`
	`  reorderQCEentries reorder_qce reorder before qce_id 2`
		Displays the QoS configuration on the controller.

Step 4	**setQCE commitcommit**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`reorderQCEentries commit`
	`ReorderQCEentries Commit Success!!!`
		Sends the QoS configuration to the NID.

Step 5	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to reorder QoS QCE on the controller:

```
Switch(config-controller-ProvisionQos)# reorderQCEentries reorder_qce qce_id 3
Switch(config-controller-ProvisionQos)# reorderQCEentries reorder reorder before qce_id 2

Switch(config-controller-ProvisionQos)# reorderQCEentries review
Commands in queue:
        reorderQCEentries reorder_qce qce_id 3
        reorderQCEentries reorder_qce qce_id 3
        reorderQCEentries reorder_qce reorder before qce_id 2
Switch(config-controller-ProvisionQos)# reorderQCEentries commit
reorderQCEentries commit
Switch(config-controller-ProvisionQos)# exit
```

Deleting QoS Control Entry (QCE) on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**deleteQCE** {**commit** | **flush** | **delete_qce** *qce-id* | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteQCE ?`
	`  commit  commit deleteQCE`
	`  delete_qce  Delete a particular QCE`
	`  flush  flush all deleteQCE commands from queue`
	`  review  review deleteQCE commands`
	`Switch(config-controller-ProvisionQos)# deleteQCE delete_qce 2`
		Deletes QoS configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **delete_qce** *qce-id***—**Deletes the QCE ID on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**deleteQCE review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteQCE review`
	`Commands in queue:`
	`  deleteQCE delete_qce 3`
		Displays the QoS configuration on the controller.

Step 3	**deleteQCE commitcommit**
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteQCE commit`
	`DeleteQCE Commit Success!!!`
		Sends the QoS configuration to the NID.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to delete QoS QCE on the controller:

```
Switch(config-controller-ProvisionQos)# deleteQCE delete_qce 2
Switch(config-controller-ProvisionQos)# deleteQCE review
Commands in queue:
        deleteQCE delete_qce 3
Switch(config-controller-ProvisionQos)# deleteQCE commit
DeleteQCE Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Deleting HQoS ID on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**deleteHqosId** {**commit** | **flush** | **delete_hqos_id** *hqos-id* | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteHqosId ?`
	`  commit  commit deleteHqosId`
	`  delete_hqos_id  deleteHqosId (default)`
	`  flush  flush all deleteHqosId commands from queue`
	`  review  review deleteHqosId commands`
	`Switch(config-controller-ProvisionQos)# deleteHqosId delete_hqos_id 2`
		Deletes HQoS ID configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **delete_hqos_id** *hqos-id*—Deletes the HQoS ID on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**deleteHqosId review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteHqosId review`
	`Commands in queue:`
	`  deleteHqosId delete_hqos_id 2`
		Displays the HQoS ID configuration on the controller.

	Command or Action	Purpose

Step 3	**deleteHqosIdcommit**
	
	Example:
	`Switch(config-controller-ProvisionQos)# deleteHqosId commit`
	`deleteHqosId Commit Success!!!`
		Sends the QoS configuration to the NID.

Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example shows how to delete HQoS IDon the controller:

```
Switch(config-controller-ProvisionQos)# deleteHqosId delete_hqos_id 2
Switch(config-controller-ProvisionQos)# deleteHqosId review
Commands in queue:
        deleteHqosId delete_hqos_id 2
Switch(config-controller-ProvisionQos)# deleteHqosId commit
DeleteHqosId Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Negating QoS and Restoring Defaults

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**no ?**
	
	Example:
	`Switch(config-controller-ProvisionQoS)# no ?`
	
	`deleteQCE Delete a particular QCE`
	`exit Exit from ProvisionQos sub configuration mode`
	`getInputGlobalPolicy Show Output QoS global features configured`
	`getInputPortPolicy Show Input Policy configured on Physical Port`
	`getOutputGlobalPolicy Show Global Output QoS features`
	`getOutputPortPolicy Show Output Policy configured on Physical Port`
	`getQCE getQCE (default)`
	`getSystemQosSettings getSystemQosSettings (default)`
	`reorderQCEentries reorderQCEentries (default)`
	`setInputGlobalPolicy configure Global Input QoS features`
	`setInputPortPolicy configure Input policy on Physical Port`
		Negates the commands and sets the default configuration.

	Command or Action	Purpose

	`setOutputGlobalPolicy  configure Global Output QoS features`
	`setOutputPortPolicy  configure Output policy on Physical Port`
	`setQCE  setQCE (default)`
	`setSystemQosSettings  set System-wide QoS settings`
	`showQCElist  showQCElist (default)`
	`showQueueStatistics  Display egress queue statistics`	

Step 2	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Viewing QoS Input Policy Global Features on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputGlobalPolicy** {**commit** | **flush** | **input** | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy ?`
	`  commit  commit getInputGlobalPolicy`
	`flush  flush all getInputGlobalPolicy commands from queue`
	`input  Show Output QoS global features configured`
	`review  review getInputGlobalPolicy commands`
	
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy input`
		View the global input QoS policy.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **input —**Displays the input policy QoS global features configured the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**getInputGlobalPolicy review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy review`
	`Commands in queue:`
	`  getInputGlobalPolicy input`
		Displays the QoS configuration on the controller.

	Command or Action	Purpose
Step 3	**getInputGlobalPolicy commit** **Example:** `Switch(config-controller-ProvisionQos)# getInputGlobalPolicy commit`	Sends the QoS configuration to the NID.
Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the QoS input port policy on the controller:

```
Switch(config-controller-ProvisionQos)# getInputGlobalPolicy input
Switch(config-controller-ProvisionQos)# getInputGlobalPolicy review
CCommands in queue:
        getInputGlobalPolicy input
Switch(config-controller-ProvisionQos)# getInputGlobalPolicy commit

GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_00.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_00.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_01.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_01.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_02.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_02.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_03.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_03.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_04.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_04.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_05.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_05.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_06.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_06.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_07.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_07.mark_egress_class
= 8
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_08.mark_DSCP = 64
GetInputGlobalPolicy_Output.inputGlobalPolicyConfig.match_DSCP.value_08.mark_egress_class
= 8
.
.
!
 GetInputGlobalPolicy Commit Success!!!


Switch(config-controller-ProvisionQos)# exit
```

Viewing QoS Input Policy Features at Port level on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputPortPolicy** {**commit** | **flush** | **input_port** *port_number*| **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputPortPolicy ?`
	`  commit  commit getInputPortPolicy`
	`  flush  flush all getInputPortPolicy commands from queue`
	`  input_port  Show Input Policy configured on Physical Port`
	`  review  review getInputPortPolicy commands`
	`Switch(config-controller-ProvisionQos)#`
	`getInputPortPolicy input_port 2`
		View the input QoS policy at port level.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **input_port** *port_number*—Displays the input port policy configuration at port level on the controller. The valid ports are 1 to 6. port 7 is invalid.
		• **review—**Displays the configuration on the controller.

Step 2	**getInputPortPolicy review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputPortPolicy review`
	
	`Commands in queue:`
	`  getInputPortPolicy input_port 3`
	`  getInputPortPolicy input_port 2`
		Displays the QoS configuration on the controller.

Step 3	**getInputPortPolicy commit**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputPortPolicy commit`
		Sends the QoS configuration to the NID.

Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the QoS input port policy on the controller:

```
Switch(config-controller-ProvisionQos)# getInputPortPolicy input_port 2
```

```
Switch(config-controller-ProvisionQos)# getInputPortPolicy review
Commands in queue:
        getInputPortPolicy input_port 3
        getInputPortPolicy input_port 2
Switch(config-controller-ProvisionQos)# getInputPortPolicy commit
GetInputPortPolicy_Output.inputPortPolicyConfig.port_number = 2
GetInputPortPolicy_Output.inputPortPolicyConfig.port_policer.state = false
GetInputPortPolicy_Output.inputPortPolicyConfig.port_policer.cir = 1000000
GetInputPortPolicy_Output.inputPortPolicyConfig.globalDscpBasedDscpIngressMarking = false
GetInputPortPolicy_Output.inputPortPolicyConfig.globalDscpBasedEgressClassMarking = false
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_0.mark_egress_class = 1
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_1.mark_egress_class = 0
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_2.mark_egress_class = 2
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_3.mark_egress_class = 3
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_4.mark_egress_class = 4
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_5.mark_egress_class = 5
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_6.mark_egress_class = 6
GetInputPortPolicy_Output.inputPortPolicyConfig.match.cos_.value_7.mark_egress_class = 7
GetInputPortPolicy_Output.inputPortPolicyConfig.egress_class_marking = false
GetInputPortPolicy_Output.inputPortPolicyConfig.qce.address.t = 1
GetInputPortPolicy_Output.inputPortPolicyConfig.qce.address.u.source = ''
GetInputPortPolicy_Output.inputPortPolicyConfig.qce.key.t = 1
GetInputPortPolicy_Output.inputPortPolicyConfig.qce.key.u.normal = ''
GetInputPortPolicy_Output.inputPortPolicyConfig.service_policy.t = 2
GetInputPortPolicy_Output.inputPortPolicyConfig.service_policy.u.detach = ''

 GetInputPortPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Viewing QoS Output Policy Global Features on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputGlobalPolicy** {**commit** | **flush** | **output** | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy ?`
	`  commit  commit getOutputGlobalPolicy`
	`  flush  flush all getOutputGlobalPolicy commands from queue`
	`  output  Show Global Output QoS features`
	`  review  review getOutputGlobalPolicy commands`
	
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy output`
		View global output QoS policy.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **input —**Displays the input policy QoS global features configured the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**getOutputPortPolicy review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getInputGlobalPolicy review`
		Displays the QoS configuration on the controller.

	Command or Action	Purpose
	`Commands in queue:` `getOutputGlobalPolicy output`	
Step 3	**getOutputPortPolicy commit** **Example:** `Switch(config-controller-ProvisionQos)#` **`getInputGlobalPolicy commit`**	Sends the QoS configuration to the NID.
Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)#` **`exit`**	Exits the QoS provisioning mode.

Configuration Example

The example displays the QoS output policy global features on the controller:

```
Switch(config-controller-ProvisionQos)# getInputGlobalPolicy output
Switch(config-controller-ProvisionQos)# getInputGlobalPolicy review
Commands in queue:
        getOutputGlobalPolicy output

Switch(config-controller-ProvisionQos)# getInputGlobalPolicy commit
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_00.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_01.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_02.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_03.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_04.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_05.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_06.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_07.mark_DSCP = 64
GetOutputGlobalPolicy_Output.outputGlobalPolicyConfig.match_DSCP.value_08.mark_D

 GetOutputGlobalPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Viewing QoS Output Policy Features at Port level on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**getOutputPortPolicy** {**commit** | **flush** | **output_port** *port_number*| **review**}	View the output Qos policy at port level. • **commit—**Sends the QoS configuration to NID.

	Command or Action	Purpose

	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy ?`
	`ccommit commit getOutputPortPolicy`
	`flush flush all getOutputPortPolicy commands from queue`
	`output_port Show Output Policy configured on Physical Port`
	`review review getOutputPortPolicy commands`
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy output_port 4`
		• **flush—**Flushes all QoS configuration from the queue.
		• **output_port** *port_number***—**Displays the output port policy configuration at port level on the controller. The valid ports are 1 to 6. port 7 is invalid.
		• **review—**Displays the configuration on the controller.

Step 2	**getOutputPortPolicy review**
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy review`
	`Commands in queue:`
	`getOutputPortPolicy output_port 4`
		Displays the QoS configuration on the controller.

Step 3	**getOutputPortPolicy commit**
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy commit`
		Sends the QoS configuration to the NID.

Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the QoS output port policy on the controller:

```
Switch(config-controller-ProvisionQos)# getOutputPortPolicy output_port 4
Switch(config-controller-ProvisionQos)# getOutputPortPolicy review
Commands in queue:
        getOutputPortPolicy output_port 4

Switch(config-controller-ProvisionQos)# getOutputPortPolicy commit
GetOutputPortPolicy_Output.outputPortPolicyConfig.port_number = 4
GetOutputPortPolicy_Output.outputPortPolicyConfig.port_shaper.state = false
GetOutputPortPolicy_Output.outputPortPolicyConfig.port_shaper.rate = 1000000
GetOutputPortPolicy_Output.outputPortPolicyConfig.globalDscpBasedDscpEgressMarking = false
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_7.bandwidth.priority_level
 = 1
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_7.shaper.state = false
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_7.shaper.rate = 1000000
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_7.mark_cos= 7
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_6.bandwidth.priority_level
 = 2
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_6.shaper.state = false
GetOutputPortPolicy_Output.outputPortPolicyConfig.match.egress_class_6.shaper.rate = 1000000
.
```

```
.
!
GetOutputPortPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Viewing QoS Control Entry (QCE) Configuration on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getQCE** {**commit** | **flush** | **QCE_ID** *qce_id*| **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy ?`
	`  QCE_ID  getQCE (default)`
	`  commit  commit getQCE`
	`  flush  flush all getQCE commands from queue`
	`  review  review getQCE commands`
	`Switch(config-controller-ProvisionQos)#`
	`getOutputPortPolicy qce-id 4`
		View the QCE configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **QCE_ID** *qce_id*—Displays the QCE configuration for QCE ID on the controller. The valid ports are 1 to 1024.
		• **review—**Displays the configuration on the controller.

Step 2	**getQCE review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# getQCE review`
	`Commands in queue:`
	`  getQCE QCE_ID 2`
	`  getQCE QCE_ID 3`
	`  getQCE QCE_ID 23`
		Displays the QoS configuration on the controller.

Step 3	**getOutputPortPolicy commit** **Example:** `Switch(config-controller-ProvisionQos)# getQCE commit`	Sends the QoS configuration to the NID.
Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the QoS output port policy on the controller:

```
Switch(config-controller-ProvisionQos)# getOutputPortPolicy qce-id 4
Switch(config-controller-ProvisionQos)# getQCE review
Commands in queue:
        getQCE QCE_ID 2
        getQCE QCE_ID 3
        getQCE QCE_ID 23
Switch(config-controller-ProvisionQos)# getQCE commit
GetQCE_Output.QCE_configuration.qce_id = 0
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_1 = false
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_2 =false
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_3 =false
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_4 =false
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_5 =false
GetQCE_Output.QCE_configuration.control.ingress_match.ports.GigabitEthernet_6 =false
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_type.t = 1
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_type.u.any =
'0'
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_fields.vlan_id_filter.t
 = 1
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_fields.vlan_id_filter.u.any
 = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_fields.cos_.t
= 1
GetQCE_Output.QCE_configuration.control.ingress_match.outer_tag_match.match_fields.cos_.u.val_any
 = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_type.t = 1
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_type.u.any =
'0'
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_fields.vlan_id_filter.t
 = 1
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_fields.vlan_id_filter.u.any
 = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_fields.inner_cos.t
 = 1
GetQCE_Output.QCE_configuration.control.ingress_match.inner_tag_match.match_fields.inner_cos.u.val_any
 = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.mac_params.smac_filter.t = 1
GetQCE_Output.QCE_configuration.control.ingress_match.mac_params.smac_filter.u.any = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.mac_params.dmac_filter.t = 1
GetQCE_Output.QCE_configuration.control.ingress_match.mac_params.dmac_filter.u.any = '0'
GetQCE_Output.QCE_configuration.control.ingress_match.frame_type.t = 1
GetQCE_Output.QCE_configuration.control.ingress_match.frame_type.u.any = '0'
GetQCE_Output.QCE_configuration.control.actions.mark_egress_class = 8
GetQCE_Output.QCE_configuration.control.actions.mark_COS = 8
GetQCE_Output.QCE_configuration.control.actions.mark_DSCP = 64

 GetQCE Commit Success!!!

GetOutputPortPolicy Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Viewing System QoS Settings on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getSystemQosSettings** {**commit** | **flush** | **QCE_ID** *qce_id*| **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getSystemQosSettings ?`
	` commit commit getSystemQosSettings`
	` flush flush all getSystemQosSettings commands from queue`
	` review review getSystemQosSettings commands`
	` system_qos getSystemQosSettings (default)`
	
	`Switch(config-controller-ProvisionQos)#`
	`getSystemQosSettings system_qos`
		View the system QoS configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **system_qos—**Displays the system QOS configuration on the controller.
		• **review—**Displays the configuration on the controller.

Step 2	**getSystemQosSettings review**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getSystemQosSettings review`
	`Commands in queue:`
	`getSystemQosSettings system_qos`
		Displays the QoS configuration on the controller.

Step 3	**getSystemQosSettings commit**
	
	Example:
	`Switch(config-controller-ProvisionQos)#`
	`getSystemQosSettings commit`
		Sends the QoS configuration to the NID.

Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the system QoS settings on the controller:

```
Switch(config-controller-ProvisionQos)# getSystemQosSettings system_qos
Switch(config-controller-ProvisionQos)# getSystemQosSettings review
Commands in queue:
        getSystemQosSettings system_qos

Switch(config-controller-ProvisionQos)# getSystemQosSettings commit
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_0.state = false
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_0.min_threshold = 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_0.max_threshold= 100
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_1.state = false
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_1.min_threshold= 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_1.max_threshold= 100
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_2.state = false
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_2.min_threshold= 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_2.max_threshold= 100
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_3.state = false
```

```
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_3.min_threshold= 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_3.max_threshold= 100
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_4.state = false
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_4.min_threshold= 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_4.max_threshold= 100
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_5.state = false
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_5.min_threshold= 0
GetSystemQosSettings_Output.system_qos_config.WRED.egress_class_5.max_threshold= 100

 GetSystemQosSettings Commit Success!!!

Switch(config-controller-ProvisionQos)# exit
```

Viewing HQoS ID on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getHqosId** {**commit** | **flush** | **hqos_id_value** *hqos_id* | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)# getHqosId ?`
	` commit  commit getHqosId`
	` flush  flush all getHqosId commands from queue`
	` hqos_id_value  getHqosId (default)`
	` review  review getHqosId commands`
	
	`Switch(config-controller-ProvisionQos)# getHqosId hqos_id_value 4`
		View the HQoS ID configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **hqos_id_value** *hqos_id*—Displays the HQOS ID configuration on the controller. The valid range is from 1 to 256.
		• **review—**Displays the configuration on the controller.

Step 2	**getHqosId review**
	
	Example:
	`Switch(config-controller-ProvisionQos)# getHqosId review`
	`Commands in queue:`
	`  getHqosId hqos_id_value 2`
		Displays the HQoS ID configuration on the controller.

Step 3	**getHqosId commit** **Example:** `Switch(config-controller-ProvisionQos)# getHqosId commit`	Sends the HQoS configuration to the NID.
Step 4	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the system HQoS ID on the controller:

```
Switch(config-controller-ProvisionQos)# getHqosId hqos_id_value 4
Switch(config-controller-ProvisionQos)# getHqosId review
Commands in queue:
        getHqosId hqos_id_value 2
Switch(config-controller-ProvisionQos)# getHqosId commit
GetHqosId_Output.hqos_id_config.hqos_id = 4
GetHqosId_Output.hqos_id_config.port_number = 4
GetHqosId_Output.hqos_id_config.shaper.state = true
GetHqosId_Output.hqos_id_config.shaper.rate = 100000
GetHqosId_Output.hqos_id_config.bandwidth.state = true
GetHqosId_Output.hqos_id_config.bandwidth.rate = 10000
GetHqosId_Output.hqos_id_config.match.egress_class_7.bandwidth.priority_level = 1
GetHqosId_Output.hqos_id_config.match.egress_class_7.shaper.state = true
GetHqosId_Output.hqos_id_config.match.egress_class_7.shaper.rate = 40000
GetHqosId_Output.hqos_id_config.match.egress_class_6.bandwidth.priority_level = 2
GetHqosId_Output.hqos_id_config.match.egress_class_6.shaper.state = true
GetHqosId_Output.hqos_id_config.match.egress_class_6.shaper.rate = 50000
GetHqosId_Output.hqos_id_config.match.egress_class_5.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_5.bandwidth.u.remaining_ratio = 5
GetHqosId_Output.hqos_id_config.match.egress_class_4.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_4.bandwidth.u.remaining_ratio = 4
GetHqosId_Output.hqos_id_config.match.egress_class_3.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_3.bandwidth.u.remaining_ratio = 4
GetHqosId_Output.hqos_id_config.match.egress_class_2.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_2.bandwidth.u.remaining_ratio = 3
GetHqosId_Output.hqos_id_config.match.egress_class_1.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_1.bandwidth.u.remaining_ratio = 3
GetHqosId_Output.hqos_id_config.match.egress_class_0.bandwidth.t = 2
GetHqosId_Output.hqos_id_config.match.egress_class_0.bandwidth.u.remaining_ratio = 2
GetHqosId Commit Success!!!

Switch(config-controller-ProvisionQos)# exit
```

Viewing EVC HQoS ID on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**getEvcHqosPolicy** {**commit** | **flush** | **evcHqosPolicy evc-id***evc_id* | **review**} **Example:** `Switch(config-controller-ProvisionQos)# getEvcHqosPolicy ?`	View the EVC HQoS ID configuration. • **commit—**Sends the QoS configuration to NID. • **flush—**Flushes all QoS configuration from the queue.

	Command or Action	Purpose
	commit commit getEvcHqosPolicy evcHqosPolicy getEvcHqosPolicy (default) flush flush all getEvcHqosPolicy commands from queue review review getEvcHqosPolicy commands Switch(config-controller-ProvisionQos)# **getEvcHqosPolicy evcHqosPolicy evc_id 1**	• **evcHqosPolicy evc-id***evc_id*—Displays the EVC HQOS ID configuration on the controller. The valid range is from 1 to 1024. • **review**—Displays the configuration on the controller.
Step 2	**getEvcHqosPolicy review** **Example:** Switch(config-controller-ProvisionQos)# **getEvcHqosPolicy review** Commands in queue: getEvcHqosPolicy evcHqosPolicy evc_id 1	Displays the EVC HQoS ID configuration on the controller.
Step 3	**getHqosId commit** **Example:** Switch(config-controller-ProvisionQos)# **getEvcHqosPolicy commit**	Sends the EVC HQoS configuration to the NID.
Step 4	**exit** **Example:** Switch(config-controller-ProvisionQos)# **exit**	Exits the QoS provisioning mode.

Configuration Example

The example displays the EVC HQoS ID on the controller:

```
Switch(config-controller-ProvisionQos)# getEvcHqosPolicy evcHqosPolicy evc_id 1
Switch(config-controller-ProvisionQos)# getEvcHqosPolicy review
Commands in queue:
        getEvcHqosPolicy evcHqosPolicy evc_id 1
Switch(config-controller-ProvisionQos)# getEvcHqosPolicy commit


Switch(config-controller-ProvisionQos)# exit
```

Displaying the QCE List on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**showQCElist** {**commit** | **flush** | **show_qce** {**all** | **specific** *specific_QCE*} | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQos)#showQCElist show_qce all`
	`Switch(config-controller-ProvisionQos)#showQCElist show_qce specific 2`
	`Switch(config-controller-ProvisionQos)#showQCElist review`
	`Switch(config-controller-ProvisionQos)#showQCElist commit`
		Displays the QCE list.
		• **show_qce—**Displays QCE list.
		• **all—**Displays entire QCE list.
		• **specific** *specific_QCE***—**Displays specific QCE list.
		• **commit—**Sends the QoS configuration to the NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **review—**Displays the QoS configuration on the controller.

Step 2	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the QCE list on the controller:

```
Switch(config-controller-ProvisionQos)#showQCElist show_qce all
Switch(config-controller-ProvisionQos)#showQCElist show_qce specific 2
Switch(config-controller-ProvisionQos)#showQCElist review
Commands in queue:
        showQCElist show_qce all
        showQCElist show_qce specific 2
Switch(config-controller-ProvisionQos)#showQCElist commit
ShowQCElist_Output.show_qce_configuration.QCE_List[0].qce_id = 2
ShowQCElist_Output.show_qce_configuration.QCE_List[0].status = false

 ShowQCElist Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```

Displaying QoS Queue Statistics on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**showQueueStatistics** {**commit** | **flush** | **queue_stats port_number** *port_num*} | **review**}
	
	Example:
	
	`Switch(config-controller-ProvisionQoS)#showQueueStatistics queue_stats port_number 3`
	`Switch(config-controller-ProvisionQoS)#showQueueStatistics review`
	`Switch(config-controller-ProvisionQoS)#showQueueStatistics commit`
		Displays the QoS queue statistics.
		• **queue_stats—**Displays egress queue statistics.
		• **port_number** *port_num***—**Displays statistics for specified port. The valid range is from 1 to 6.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS from the queue.
		• **review—**Displays the QoS configuration on the controller.

Step 2	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the egress queue statistics on the controller:

```
Switch(config-controller-ProvisionQoS)#showQueueStatistics queue_stats port_number 3
Switch(config-controller-ProvisionQoS)#showQueueStatistics review
Commands in queue:
        showQueueStatistics queue_stats port_number 3

Switch(config-controller-ProvisionQoS)#showQueueStatistics commit
ShowQueueStatistics_Output.queue_statistics.port_number = 3
ShowQueueStatistics_Output.queue_statistics.Queue_0.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_1.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_2.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_3.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_4.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_5.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_6.frames = 0
ShowQueueStatistics_Output.queue_statistics.Queue_7.frames = 0

ShowQueueStatistics Commit Success!!!
Switch(config-controller-ProvisionQoS)# exit
```

Displaying the Hierarchical QoS ID List on the Controller

Before You Begin

- Perform the steps to provision QoS on the controller. See Provisioning the Controller to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**showHqosIdList** {**commit** | **flush** | **show_hqos_id** {**all** | **specific** *specific_QCE*} | **review**}
	
	Example:
	`Switch(config-controller-ProvisionQos)#showHqosIdList show_hqos_id all`
	`Switch(config-controller-ProvisionQos)#showHqosIdList show_hqos_id specific 2`
	`Switch(config-controller-ProvisionQos)#showHqosIdList review`
	`Switch(config-controller-ProvisionQos)#showHqosIdList commit`
		Displays the HQoS ID list.
		• **show_hqos_id—**Displays HQoS ID list.
		• **all—**Displays entire HQoS ID list.
		• **specific** *specific_hqos-id***—**Displays specific HQos ID list.
		• **commit—**Sends the QoS configuration to the NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **review—**Displays the QoS configuration on the controller.

Step 2	**exit** **Example:** `Switch(config-controller-ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The example displays the HQoS ID list on the controller:

```
Switch(config-controller-ProvisionQos)#showHqosIdList show_hqos_id specific 2
Switch(config-controller-ProvisionQos)#showHqosIdList review
Commands in queue:
        showHqosIdList show_hqos_id all
        showHqosIdList show_hqos_id specific 2
Switch(config-controller-ProvisionQos)#showHqosIdList commit
ShowHqosIdList_Output.show_hqos_id_response.hqos_id_list[0].hqos_id = 2
ShowHqosIdList_Output.show_hqos_id_response.hqos_id_list[0].status = false

ShowHqosIdList Commit Success!!!
Switch(config-controller-ProvisionQos)# exit
```