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 ME 1200 NID to Configure QoS

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ProvisionQos** **Example:** `Switch# ProvisionQos`	Enters QoS provisioning mode.

Step 2	**ProvisionQos** {**default** | **deleteQCE** | **exit** | **getInputGlobalPolicy** | **getInputGlobalPolicy** | **getOutputGlobalPolicy** | **getOutputPortPolicy** | **getQCE** | **getSystemQosSettings** | **no** | **reorderQCEentries** | **setInputGlobalPolicy** | **setInputPortPolicy** | **setOutputGlobalPolicy** | **setOutputPortPolicy** | **setQCE** | **setSystemQosSettings** | **showQCElist** | **showQueueStatistics**}
	
	Example:
	`Switch(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.

Step 3	**exit** **Example:** `Switch(ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

The following example shows the supported QoS configuration:

```
Switch(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 ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setInputGlobalPolicy** {**commit** | **flush** | **inputGlobalPolicyConfig** | **review**}
	
	Example:
	
	`Switch(ProvisionQos)#`
	`setInputGlobalPolicy ?`
	`  commit  commit setInputGlobalPolicy`
	`  flush  flush all setInputGlobalPolicy commands from queue`
	`  inputGlobalPolicyConfig  configure Global Input QoS features`
	`  review  review setInputGlobalPolicy commands`
		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 NID.
		• **review—**Displays the configuration on the NID.

Step 2	**setInputGlobalPolicy inputGlobalPolicyConfig** {**match-DSCP value_xx** {**mark-DSCP** *dscp-value* | **mark-egress-class** *egress-queue*} | **service-policy** {**attach** | **detach**}}
	
	Example:
	
	`Switch(ProvisionQos)#`
	`setInputGlobalPolicy inputGlobalPolicyConfig match-DSCP`
		Configures input global policy.
		• **match-DSCP—**Matches a particular DSCP value .
		• **value-xx—**Sets DSCP values on the ME 1200 NID. Use the following values:
		• **value_00** DSCP 0. Default PHB for best effort traffic
		value-01 DSCP 1
		value-02 DSCP 2

Command or Action	Purpose
value-00 mark-DSCP 4 Switch(ProvisionQos)# **setInputGlobalPolicy inputGlobalPolicyConfig match-DSCP value-00 mark-egress-class 4** Switch(ProvisionQos)# **setInputGlobalPolicy inputGlobalPolicyConfig service-policy attach**	**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 **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

	Command or Action	Purpose
		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 **value-60** DSCP 60 **value-61** DSCP 61 **value-62** DSCP 62 **value-63** DSCP 63 • **mark-DSCP—**Marks the DSCP on the ME 1200 NID. 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.

	Command or Action	Purpose
		◦ **detach—**Removes the service policy, removes all configuration and restore the default configuration.

Step 3	**setInputGlobalPolicy review**
	
	Example:
	`Switch(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 NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig match-DSCP value-00
mark-DSCP 4
Switch(ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig match-DSCP value-00
mark-egress-class 4
Switch(ProvisionQos)# setInputGlobalPolicy inputGlobalPolicyConfig service-policy attach
Switch(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(ProvisionQos)# setInputGlobalPolicy commit
SetInputGlobalPolicy Commit Success!!!
Switch(ProvisionQos)# exit
```

Configuring QoS Input Policy Features at Port level on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setInputPortPolicy** {**commit** | **flush** | **inputPortPolicyConfig** | **review**}
	
	Example:
	
	`Switch(ProvisionQos)# setInputPortPolicy ?`
	`  commit  commit setInputPortPolicy`
	`  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 NID.
		• **review—**Displays the configuration on the NID.

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**}}
	
	Example:
	
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig egress-class-marking enable`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig globalDscpBasedDscpIngressMarking enable`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig globalDscpBasedEgressClassMarking enable`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig match cos value_0`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-number 1`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-policer cir 1000`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-policer state enable`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce address destination`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce key double-tag`
	`Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig service-policy attach`
		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.
		• **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.

	Command or Action	Purpose
		◦ **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. ◦ **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.

	Command or Action	Purpose

Step 3	**setInputPortPolicy review**
	
	Example:
	`Switch(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 NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig egress-class-marking enable
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
globalDscpBasedDscpIngressMarking enable
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig
globalDscpBasedEgressClassMarking enable
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig match cos value_0
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-number 1
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-policer cir 1000
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig port-policer state enable
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce address destination
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig qce key double-tag
Switch(ProvisionQos)# setInputPortPolicy inputPortPolicyConfig service-policy attach
Switch(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(ProvisionQos)# setInputPortPolicy commit
SetInputPortPolicy Commit Success!!!
Switch(ProvisionQos)# exit
```

Configuring QoS Output Policy Features Globally on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setOutputGlobalPolicy** {**commit** | **flush** | **outputGlobalPolicyConfig** | **review**}
	
	Example:
	
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

Step 2	**setOutputGlobalPolicy outputGlobalPolicyConfig** {**match-DSCP value-xx** {**mark-DSCP** *dscp-value*} | **service-policy** {**attach** | **detach**}}
	
	Example:
	
	`Switch(ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig match-DSCP value-01 mark-DSCP 1`
	`Switch(ProvisionQos)# setOutputGlobalPolicy`
		Configures output global policy.
		• **match-DSCP—**Matches a particular DSCP value .
		• **value-xx—**Sets DSCP values on the NID. 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

	Command or Action	Purpose
	`outputGlobalPolicyConfig service-policy attach`	**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 **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

	Command or Action	Purpose
		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 **value-60** DSCP 60 **value-61** DSCP 61 **value-62** DSCP 62 **value-63** DSCP 63 • **mark-DSCP—**Marks the DSCP on the NID. 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.

	Command or Action	Purpose

Step 3	**setOutputGlobalPolicy review**
	
	Example:
	`Switch(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 NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig match-DSCP value-00
mark-DSCP 4
Switch(ProvisionQos)# setOutputGlobalPolicy outputGlobalPolicyConfig service-policy attach
Switch(ProvisionQos)# setOutputGlobalPolicy review
Commands in queue:
        setOutputGlobalPolicy outputGlobalPolicyConfig match-DSCP value-01 mark-DSCP 1
        setOutputGlobalPolicy outputGlobalPolicyConfig service-policy attach

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

Configuring QoS Output Policy Features at Port level on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setOutputPortPolicy** {**commit** | **flush** | **outputPortPolicyConfig** | **review**}
	
	Example:
	
	`Switch(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 NID.
		
		• **review—**Displays the configuration on the NID.

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(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig cos-marking enable`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig globalDscpBasedDscpEgressMarking enable`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0 bandwidth priority-level 8`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0 bandwidth remaining-ratio 20`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0 mark-cos 7`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-number 1`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-shaper rate 1000`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-shaper state enable`
	`Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig service-policy attach`
		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.
		
		◦ **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.

	Command or Action	Purpose
		• **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(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`
		Displays the QoS configuration on the NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig cos-marking enable
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig
globalDscpBasedDscpEgressMarking enable
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0
bandwidth priority-level 8
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0
bandwidth remaining-ratio 20
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig match egress-class-0
mark-cos 7
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-number 1
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-shaper rate 1000
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig port-shaper state enable
Switch(ProvisionQos)# setOutputPortPolicy outputPortPolicyConfig service-policy attach
Switch(ProvisionQos)# setOutputPortPolicy review
Commands in queue:
        setOutputPortPolicy outputPortPolicyConfig cos-marking enable
        setOutputPortPolicy outputPortPolicyConfig globalDscpBasedDscpEgressMarking enable
        setOutputPortPolicy outputPortPolicyConfig match egress-clas-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(ProvisionQos)# setOutputPortPolicy commit
SetInputPortPolicy Commit Success!!!
Switch(ProvisionQos)# exit
```

Setting Default QoS Configuration on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**default ?**
	
	Example:
	`Switch(ProvisionQoS)# default ?`
	
	`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`
	`setOutputGlobalPolicy configure Global Output QoS features`
	`setOutputPortPolicy configure Output policy on Physical Port`
		Sets the default QoS configuration.

	Command or Action	Purpose

	`setQCE setQCE (default)`
	`setSystemQosSettings set System-wide QoS settings`
	`showQCElist showQCElist (default)`
	`showQueueStatistics Display egress queue statistics`	

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

Configuring QoS Control Entry (QCE) on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setQCE** {**commit** | **flush** | **QCE-configuration** | **review**}
	
	Example:
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

Step 2	**setQCE QCE-configuration** {**control** {**actions** | **ingress-match**} | **qce-id**}
	
	Example:
	`Switch(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress-match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, on page 20.
		• **qce-id—**Specifies the QCE ID. The valid range is from 1 to 1024. 0 is invalid.

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

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4
Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields
inner-cos val-0-1
Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan
 c-tagged
Switch(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(ProvisionQos)# setQCE commit
 SetQCE Commit Success!!!
Switch(ProvisionQos)# exit
```

Configuring QoS Control Entry (QCE) Control Actions on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**setQCE QCE-configuration** {**control** {**actions** | **ingress-match**} | **qce-id**}	Configures QCE.

	Command or Action	Purpose

	Example:
	`Switch(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(ProvisionQos)# setQCE QCEconfiguration qce-id 4`
		• **control—**Configures QCE.
		• **actions—**Configures QCE actions. See Configuring QoS Control Entry (QCE) Control Actions on the ME 1200 NID, on page 18.
		• **ingress-match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, on page 20.
		• **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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration control action mark-DSCP 3`
	`Switch(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(ProvisionQos)# setQCE review`	Displays the QoS configuration on the NID.
Step 4	**setQCE commit** **Example:** `Switch(ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(ProvisionQos)# exit`	Exits the QoS provisioning mode.

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCEConfig match-DSCP value-00 mark-DSCP 4
Switch(ProvisionQos)# setQCE QCEConfig match-DSCP value-00 mark-egress-class 4
Switch(ProvisionQos)# setQCE QCEConfig service-policy attach
Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Match Ingress Parameters on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		
		• **ingress-match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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(ProvisionQos)# setQCE QCE-configuration control ingress-match frame-type any match-fields inner-cos val-0-1 any`
	`Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match inner-tag-match match-fields inner-cos val-0-3`
	`Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match mac-params dmac-filter any`
	`Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match outer-tag_match match-fields cos val-2-3`
	`Switch(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 ME 1200 NID, on page 21.
		
		◦ **inner-tag-match—**Matches against inner tag. See Configuring QCE Control Ingress Inner Tag Match Parameter on the ME 1200 NID, on page 24.
		
		◦ **mac-params—**Matches against MAC filters. See Configuring QCE Control Ingress MAC Params Parameter on the ME 1200 NID, on page 27.
		
		◦ **outer-tag-match—**Matches against outer tag. See Configuring QCE Control Ingress Outer Tag Match Parameter on the ME 1200 NID, on page 28.
		
		◦ **ports—**Matches against ports. See Configuring QCE Control Ingress Ports Parameter on the ME 1200 NID, on page 31.

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

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match frame-type any
match-fields inner-cos val-0-1 any
Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match inner-tag-match
match-fields inner-cos val-0-3
Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match mac-params dmac-filter
 any
Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match outer-tag-match
match-fields cos val-2-3
Switch(ProvisionQos)# setQCE QCE-configuration control ingress-match ports gigabitEthernet-2
 enable
Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Control Ingress Match Frame Type Parameter on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress-match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter host host1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter network dest-ip-addr addr2`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dscp-filter host any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dscp-filter range 3-4`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 fragment-type frag`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 protocol specific 45`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 source-ip-filter network source-ip-mask soumask`
	`Switch(ProvisionQos)# setQCE QCE-configuration`
		• **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
	ingress-match frame-type ipv6 dest-ip-filter any Switch(ProvisionQos)# **setQCE QCE-configuration ingress-match frame-type ipv6 dest-ip-filter specfic dest-ip-addr-32its dest34** Switch(ProvisionQos)# **setQCE QCE-configuration ingress-match frame-type ipv6 dscp-filter specific 45 any** Switch(ProvisionQos)# **setQCE QCE-configuration ingress-match frame-type ipv6 protocol specific 450** Switch(ProvisionQos)# **setQCE QCE-configuration ingress-match frame-type ipv6 protocol specific 45** Switch(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(ProvisionQos)# **setQCE review**	Displays the QoS configuration on the NID.
Step 4	**setQCE commit** **Example:** Switch(ProvisionQos)# **setQCE commit**	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** Switch(ProvisionQos)# **exit**	Exits the QoS provisioning mode.

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter
 any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter
 host host1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dest-ip-filter
 network dest-ip-addr addr2
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dscp-filter
host any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 dscp-filter
range 3-4
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 fragment-type
 frag
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 protocol
specific 45
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv4 source-ip-filter
 network source-ip-mask soumask
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 dest-ip-filter
 any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 dest-ip-filter
 specfic dest-ip-addr-32its dest34
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 dscp-filter
specific 45 any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 protocol
specific 450
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 protocol
specific 45
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match frame-type ipv6 source-ip-filter
 specific source-ip-mask source-mask

Switch(ProvisionQos)# setQCE review
Commands in queue:
       setQCE QC-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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Control Ingress Inner Tag Match Parameter on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress-match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields inner-cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields vlan-id-filter any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields vlan-id-filter range range1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields vlan-id-filter specific 450`
	`Switch(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.
		◦ **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.

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

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
inner-cos val-0-1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter range range1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter specific 450
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-type
c-tagged

Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Control Ingress MAC Params Parameter on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner_cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress_match**—Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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** *specific-fillter* | **unicast**} |**smac-filter** {**any** | **specific** *specific-filter*}
	
	Example:
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match mac-params dmac-filter any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match mac-params dmac-filter broadcast`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match mac-params dmac-filter specific filter1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match mac-params smac-filter any`
		• **control**—Configures QCE.
		• **ingress-match**—Configures ingress match.
		• **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.

	Command or Action	Purpose
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match mac-params smac-filter specific filter2`	
Step 3	**setQCE review** **Example:** `Switch(ProvisionQos)# setQCE review`	Displays the QoS configuration.
Step 4	**setQCE commit** **Example:** `Switch(ProvisionQos)# setQCE commit`	Sends the QoS configuration to the NID.
Step 5	**exit** **Example:** `Switch(ProvisionQos)#exit`	Exits the QoS mode.

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
inner-cos val-0-1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter range range1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-fields
vlan-id-filter specific 450
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match inner-tag-match match-type
c-tagged

Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Control Ingress Outer Tag Match Parameter on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner_cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress_match—**Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields vlan-id-filter any`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields vlan-id-filter range range1`
	`Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields vlan-id-filter specific 230`
	`Switch(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.
		◦ **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.

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

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields
cos val-0-1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields
vlan-id-filter any
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields
vlan-id-filter range range1
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-fields
vlan-id-filter specific 230
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match outer-tag-match match-type
c-tagged

Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring QCE Control Ingress Ports Parameter on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# setQCE QCEconfiguration control action mark-Cos 4`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-fields inner_cos val-0-1`
	`Switch(ProvisionQos)# setQCE QCEconfiguration ingress-match frame-type any match-type vlan c-tagged`
	`Switch(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 ME 1200 NID, on page 18.
		• **ingress_match**—Configures ingress match. See Configuring QCE Match Ingress Parameters on the ME 1200 NID, 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(ProvisionQos)# setQCE QCE-configuration ingress-match ports GigabitEthernet-1 enable`
	`Switch(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.

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

Configuration Example

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

```
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match ports GigabitEthernet-1 enable
Switch(ProvisionQos)# setQCE QCE-configuration ingress-match ports GigabitEthernet-3 disable

Switch(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(ProvisionQos)# setQCE commit
Switch(ProvisionQos)# exit
```

Configuring System QoS on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setSystemQosSettings** {**commit** | **flush** | **system-qos-config** | **review**}
	
	Example:
	`Switch(ProvisionQos)# setSystemQosSettings ?`
	`  commit  commit setSystemQosSettings`
		Sets the system Qos configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.

	Command or Action	Purpose
	flush flush all setSystemQosSettings commands from queue review review setSystemQosSettings commands system-qos-config set System-wide QoS settings	• **system-qos-config—**Sets the system wide QoS settings on the NID. • **review—**Displays the configuration on the NID.

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(ProvisionQos)# setSystemQosSettings system-qos-config WRED egress-class-0 max-threshold 20`
	`Switch(ProvisionQos)# setSystemQosSettings system-qos-config WRED egress-class-1 min-threshold 40`
	`Switch(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.
		• **enable—**Enables the WRED.
		• **disable—**Disables the WRED.

Step 3	**setSystemQosSettings review**
	
	Example:
	`Switch(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 NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setSystemQosSettings system-qos-config WRED egress-class-0 max-threshold
 20
Switch(ProvisionQos)# setSystemQosSettings system-qos-config WRED egress-class-1 min-threshold
 40
Switch(ProvisionQos)# setSystemQosSettings system-qos-config WRED egress-class-2 state
enable
Switch(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(ProvisionQos)# setSystemQosSettings commit
Switch(ProvisionQos)# exit
```

Configuring Hierarchical QoS on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setsetHqosId** {**commit** | **flush** | **hqos-id-config** | **review**}
	
	Example:
	
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

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(ProvisionQos)# sethqosid hqos-id-config`	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.

	Command or Action	Purpose
	bandwidth rate 100 Switch(ProvisionQos)# **sethqosid hqos-id-config bandwidth state enable** Switch(ProvisionQos)# **sethqosid hqos-id-config hqos-id 4** Switch(ProvisionQos)# **sethqosid hqos-id-config match egress-class-7 bandwidth priority-level 1** Switch(ProvisionQos)# **sethqosid hqos-id-config match egress-class-7 shaper rate 100** Switch(ProvisionQos)# **sethqosid hqos-id-config match egress-class-7 shaper state enable** Switch(ProvisionQos)# **sethqosid hqos-id-config port 2** Switch(ProvisionQos)# **sethqosid hqos-id-config shaper rate 100** Switch(ProvisionQos)# **sethqosid hqos-id-config shaper state enable**	• **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. • **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(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 NID.
Step 4	**setHqosIdcommit** **Example:** Switch(ProvisionQos)# **setHqosId commit**	Sends the QoS configuration to the NID.

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

Configuration Example

The example shows how to configure HQOS on the NID:

```
Switch(ProvisionQos)# sethqosid hqos-id-config bandwidth rate 100
Switch(ProvisionQos)# sethqosid hqos-id-config bandwidth state enable
Switch(ProvisionQos)# sethqosid hqos-id-config hqos-id 4
Switch(ProvisionQos)# sethqosid hqos-id-config match egress-class-7 bandwidth priority-level
 1
Switch(ProvisionQos)# sethqosid hqos-id-config match egress-class-7 shaper rate 100
Switch(ProvisionQos)# sethqosid hqos-id-config match egress-class-7 shaper state enable
Switch(ProvisionQos)# sethqosid hqos-id-config port 2
Switch(ProvisionQos)# sethqosid hqos-id-config shaper rate 100
Switch(ProvisionQos)# sethqosid hqos-id-config shaper state enable
Switch(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(ProvisionQos)# setHqosId commit
SetHqosId Commit Success!!!
Switch(ProvisionQos)# exit
```

Configuring EVC Hierarchical QoS Policy on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose
	flush flush all setEvcHqosPolicy commands from queue review review setEvcHqosPolicy commands	• **evcHqosPolicyConfig—**Sets the EVC HQoS policy on the NID. • **review—**Displays the configuration on the NID.

Step 2	**setEvcHqosPolicy evcHqosPolicyConfig** {**evc-id** *evc-id* | **hqos-id** *hqos-id* | **service-policy** {**attach** | **detach**}
	Example:
	`Switch(ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig evc-id 1`
	`Switch(ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig hqos-id 2`
	`Switch(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.

Step 3	**setEvcHqosPolicy review**
	Example:
	`Switch(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 NID.

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

Configuration Example

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

```
Switch(ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig evc-id 1
Switch(ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig hqos-id 2
Switch(ProvisionQos)# setEvcHqosPolicy evcHqosPolicyConfig attach
Switch(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(ProvisionQos)# setEvcHqosPolicy commit
Switch(ProvisionQos)# exit
```

Reordering QoS Control Entry (QCE) on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**reorderQCEentries** {**commit** | **flush** | **reorder-qce** | **review**}
	
	Example:
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

Step 2	**reorderQCEentries reorder-qce** {**qce-id** *qce-id* | **reorder** {**after** | **before** | **last**} {**qce-id** *qce-id*}}
	
	Example:
	`Switch(ProvisionQos)# reorderQCEentries reorder-qce qce-id 3`
	`Switch(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(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 NID.

	Command or Action	Purpose

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

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

Configuration Example

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

```
Switch(ProvisionQos)# reorderQCEentries reorder-qce qce-id 3
Switch(ProvisionQos)# reorderQCEentries reorder reorder before qce-id 2

Switch(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(ProvisionQos)# reorderQCEentries commit
reorderQCEentries commit
Switch(ProvisionQos)# exit
```

Deleting QoS Control Entry (QCE) on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**deleteQCE** {**commit** | **flush** | **delete-qce** *qce-id* | **review**}
	
	Example:
	`Switch(ProvisionQos)# deleteQCE ?`
	`  commit  commit deleteQCE`
	`  delete-qce  Delete a particular QCE`
	`  flush  flush all deleteQCE commands from queue`
	`  review  review deleteQCE commands`
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

	Command or Action	Purpose

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

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

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

Configuration Example

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

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

Deleting HQoS ID on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# deleteHqosId ?`
	`commit commit deleteHqosId`
	`delete-hqos-id deleteHqosId (default)`
		Deletes HQoS ID configuration.
		• **commit**—Sends the QoS configuration to NID.
		• **flush**—Flushes all QoS configuration from the queue.

	Command or Action	Purpose
	flush flush all deleteHqosId commands from queue review review deleteHqosId commands Switch(ProvisionQos)# **deleteHqosId delete-hqos-id 2**	• **delete-hqos-id** *hqos-id*—Deletes the HQoS ID on the NID. • **review**—Displays the configuration on the NID.
Step 2	**deleteHqosId review** **Example:** Switch(ProvisionQos)# **deleteHqosId review** Commands in queue: deleteHqosId delete-hqos-id 2	Displays the HQoS ID configuration on the NID.
Step 3	**deleteHqosIdcommit** **Example:** Switch(ProvisionQos)# **deleteHqosId commit** deleteHqosId Commit Success!!!	Sends the QoS configuration to the NID.
Step 4	**exit** **Example:** Switch(ProvisionQos)#**exit**	Exits the QoS mode.

Configuration Example

The example shows how to delete HQoS IDon the NID:

```
Switch(ProvisionQos)# deleteHqosId delete-hqos-id 2
Switch(ProvisionQos)# deleteHqosId review
Commands in queue:
        deleteHqosId delete-hqos-id 2
Switch(ProvisionQos)# deleteHqosId commit
DeleteHqosId Commit Success!!!
Switch(ProvisionQos)# exit
```

Negating QoS and Restoring Defaults

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**no ?**
	
	Example:
	`Switch(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`
	`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`
		Negates the commands and sets the default configuration.

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

Viewing QoS Input Policy Global Features on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputGlobalPolicy** {**commit** | **flush** | **input** | **review**}
	
	Example:
	
	`Switch(ProvisionQos)# getInputGlobalPolicy ?`
	`commit commit getInputGlobalPolicy`
	`flush flush all getInputGlobalPolicy commands from queue`
	`input Show Output QoS global features configured`
	`review review getInputGlobalPolicy commands`
		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 NID.

	Command or Action	Purpose
	`Switch(ProvisionQos)# getInputGlobalPolicy input`	• **review—**Displays the configuration on the NID.

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

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

Configuration Example

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

```
Switch(ProvisionQos)# getInputGlobalPolicy input
Switch(ProvisionQos)# getInputGlobalPolicy review
CCommands in queue:
        getInputGlobalPolicy input
Switch(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(ProvisionQos)# exit
```

Viewing QoS Input Policy Features at Port level on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputPortPolicy** {**commit** | **flush** | **input-port** *port-number*| **review**}
	
	Example:
	`Switch(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(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 NID. The valid ports are 1 to 6. port 7 is invalid.
		• **review—**Displays the configuration on the NID.

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

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

Configuration Example

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

```
Switch(ProvisionQos)# getInputPortPolicy input_port 2
Switch(ProvisionQos)# getInputPortPolicy review
Commands in queue:
        getInputPortPolicy input_port 3
        getInputPortPolicy input_port 2
Switch(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(ProvisionQos)# exit
```

Viewing QoS Output Policy Global Features on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getInputGlobalPolicy** {**commit** | **flush** | **output** | **review**}
	
	Example:
	
	`Switch(ProvisionQos)# getInputGlobalPolicy ?`
	`  commit  commit getOutputGlobalPolicy`
	`  flush  flush all getOutputGlobalPolicy commands from queue`
	`  output  Show Global Output QoS features`
	`  review  review getOutputGlobalPolicy commands`
	
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

	Command or Action	Purpose

Step 2	**getOutputPortPolicy review**
	
	Example:
	`Switch(ProvisionQos)# getInputGlobalPolicy review`
	`Commands in queue:`
	`getOutputGlobalPolicy output`
		Displays the QoS configuration on the NID.

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

Configuration Example

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

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

Switch(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(ProvisionQos)# exit
```

Viewing QoS Output Policy Features at Port level on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getOutputPortPolicy** {**commit** | **flush** | **output-port** *port-number*| **review**}
	
	Example:
	`Switch(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(ProvisionQos)# getOutputPortPolicy output-port 4`
		View the output Qos policy at port level.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **output-port** *port-number***—**Displays the output port policy configuration at port level on the NID. The valid ports are 1 to 6. port 7 is invalid.
		• **review—**Displays the configuration on the NID.

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

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

Configuration Example

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

```
Switch(ProvisionQos)# getOutputPortPolicy output-port 4
Switch(ProvisionQos)# getOutputPortPolicy review
Commands in queue:
        getOutputPortPolicy output-port 4

Switch(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(ProvisionQos)# exit
```

Viewing QoS Control Entry (QCE) Configuration on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

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

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

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

Configuration Example

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

```
Switch(ProvisionQos)# getOutputPortPolicy qce-id 4
Switch(ProvisionQos)# getQCE review
Commands in queue:
        getQCE QCE-ID 2
        getQCE QCE-ID 3
        getQCE QCE-ID 23
Switch(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(ProvisionQos)# exit
```

Viewing System QoS Settings on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID to Configure QoS, on page 2.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getSystemQosSettings** {**commit** | **flush** | **QCE-ID** *qce-id*| **review**}
	
	Example:
	`Switch(ProvisionQos)# getSystemQosSettings ?`
	` commit commit getSystemQosSettings`
	` flush flush all getSystemQosSettings commands from queue`
	` review review getSystemQosSettings commands`
	` system-qos getSystemQosSettings (default)`
	
	`Switch(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 NID.
		• **review—**Displays the configuration on the NID.

Step 2	**getSystemQosSettings review**
	
	Example:
	`Switch(ProvisionQos)# getSystemQosSettings review`
	`Commands in queue:`
	`getSystemQosSettings system-qos`
		Displays the QoS configuration on the NID.

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

Configuration Example

The example displays the system QoS settings on the NID:

```
Switch(ProvisionQos)# getSystemQosSettings system-qos
Switch(ProvisionQos)# getSystemQosSettings review
Commands in queue:
        getSystemQosSettings system-qos

Switch(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(ProvisionQos)# exit
```

Viewing EVC HQoS ID on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# getEvcHqosPolicy ?`
	`commit commit getEvcHqosPolicy`
	`evcHqosPolicy getEvcHqosPolicy (default)`
	`flush flush all getEvcHqosPolicy commands from queue`
	`review review getEvcHqosPolicy commands`
	
	`Switch(ProvisionQos)# getEvcHqosPolicy evcHqosPolicy evc-id 1`
		View the EVC HQoS ID configuration.
		• **commit—**Sends the QoS configuration to NID.
		• **flush—**Flushes all QoS configuration from the queue.
		• **evcHqosPolicy evc-id***evc-id***—**Displays the EVC HQOS ID configuration on the NID. The valid range is from 1 to 1024.
		• **review—**Displays the configuration on the NID.

Step 2	**getEvcHqosPolicy review**
	
	Example:
	`Switch(ProvisionQos)# getEvcHqosPolicy review`
	`Commands in queue:`
	`getEvcHqosPolicy evcHqosPolicy evc-id 1`
		Displays the EVC HQoS ID configuration on the NID.

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

Configuration Example

The example displays the EVC HQoS ID on the NID:

```
Switch(ProvisionQos)# getEvcHqosPolicy evcHqosPolicy evc-id 1
Switch(ProvisionQos)# getEvcHqosPolicy review
Commands in queue:
        getEvcHqosPolicy evcHqosPolicy evc-id 1
Switch(ProvisionQos)# getEvcHqosPolicy commit


Switch(ProvisionQos)# exit
```

Viewing HQoS ID on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)# getHqosId ?`
	` commit  commit getHqosId`
	` flush  flush all getHqosId commands from queue`
	` hqos-id-value  getHqosId (default)`
	` review  review getHqosId commands`
	
	`Switch(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 NID. The valid range is from 1 to 256.
		• **review—**Displays the configuration on the NID.

Step 2	**getHqosId review**
	
	Example:
	`Switch(ProvisionQos)# getHqosId review`
	`Commands in queue:`
	`  getHqosId hqos-id-value 2`
		Displays the HQoS ID configuration on the NID.

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

Configuration Example

The example displays the system HQoS ID on the NID:

```
Switch(ProvisionQos)# getHqosId hqos-id-value 4
Switch(ProvisionQos)# getHqosId review
Commands in queue:
        getHqosId hqos-id-value 2
Switch(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(ProvisionQos)# exit
```

Displaying the Hierarchical QoS ID List on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)#showHqosIdList show-hqos-id all`
	`Switch(ProvisionQos)#showHqosIdList show-hqos-id specific 2`
	`Switch(ProvisionQos)#showHqosIdList review`
	`Switch(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.

	Command or Action	Purpose
		• **review—**Displays the QoS configuration on the NID.
Step 2	**exit** **Example:** `Switch(ProvisionQos)#exit`	Exits the QoS mode.

Configuration Example

The example displays the HQoS ID list on the NID:

```
Switch(ProvisionQos)#showHqosIdList show-hqos-id specific 2
Switch(ProvisionQos)#showHqosIdList review
Commands in queue:
        showHqosIdList show-hqos-id all
        showHqosIdList show-hqos-id specific 2
Switch(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(ProvisionQos)# exit
```

Displaying the QCE List on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQos)#showQCElist show-qce all`
	`Switch(ProvisionQos)#showQCElist show-qce specific 2`
	`Switch(ProvisionQos)#showQCElist review`
	`Switch(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 NID.

	Command or Action	Purpose
Step 2	**exit** **Example:** `Switch(ProvisionQos)#exit`	Exits the QoS mode.

Configuration Example

The example displays the QCE list on the NID:

```
Switch(ProvisionQos)#showQCElist show_qce all
Switch(ProvisionQos)#showQCElist show_qce specific 2
Switch(ProvisionQos)#showQCElist review
Commands in queue:
        showQCElist show-qce all
        showQCElist show-qce specific 2
Switch(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(ProvisionQos)# exit
```

Displaying QoS Queue Statistics on the ME 1200 NID

Before You Begin

- Perform the steps to provision QoS on the ME 1200 NID. See Provisioning the ME 1200 NID 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(ProvisionQoS)#showQueueStatistics queue-stats port-number 3`
	`Switch(ProvisionQoS)#showQueueStatistics review`
	`Switch(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 NID.

	Command or Action	Purpose
Step 2	**exit** **Example:** `Switch(ProvisionQos)#exit`	Exits the QoS mode.

Configuration Example

The example displays the egress queue statistics on the NID:

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

Switch(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(ProvisionQoS)# exit
```