Modular QoS command-line interface

Modular QoS command-line interface (MQC) is a Quality of Service (QoS) CLI configuration framework that

- defines traffic classification rules using class-map objects
- associates classification with treatment actions for traffic packets in policy-map objects, and
- attaches traffic policies to interfaces (ingress or egress), thereby enabling consistent QoS control across platforms.

MQC enables users to create QoS policies by linking defined traffic classes with specific traffic treatment actions (marking, policing, shaping, queuing, or dropping) and then applying these policies to interfaces. The system enforces classification and subsequent treatment on packets flowing through the interfaces and enforces the QoS behavior described in the associated policy map.

MQC comprises of three fundamental logical components.

- Class-map: defines a traffic class used later by a policy-map as a match criteria to apply QoS treatment.
- Policy-map: associates one or more traffic classes with QoS actions such as marking, queueing, scheduling, shaping, WRED, or policing.
- Service-policy: applies a policy-map to an interface (ingress or egress), activating the configured QoS treatments for matching traffic.

MQC attributes

MQC exhibits several structural characteristics that define how it organizes and applies QoS policies.

- Implements a three-stage object model: classification (class-map), policy creation (policy-map), and policy attachment to interfaces (service-policy).
- Enables reuse of named objects (classes and policies) to reduce duplicated configuration.
- Distinguishes classification logic from traffic treatment logic so that traffic definition and QoS action remain decoupled.
- Includes default handling: traffic that doesn't match any class is bound to a default class with baseline behavior.
- Supports nested class-maps using match class-map to refine matching hierarchy.

Table 1: Differences between MQC and traditional QoS CLI frameworks

Attribute	MQC framework	Traditional QoS CLI framework
Type	Configuration framework consists of modular command hierarchy	Interface-centric command model consists of flat or monolithic CLI
Structure	Uses modular objects: class-map, policy-map, and service-policy	Defines QoS directly under each interface with feature-specific commands
Reusability	High—classes and policies are reusable across interfaces and services	Low—each interface must be configured independently
Consistency	Ensures identical QoS behavior across multiple interfaces	Risk of inconsistency due to repetitive manual configuration
Maintainability	Simplifies updates—change one policy and re-apply	Requires per-interface configuration changes for every modification

Benefits of MQC

MQC offers multiple operational advantages that enhance the consistency and scalability of QoS configuration.

- Promotes consistency in QoS configurations by using the same architecture on multiple platforms.
- Improves readability and maintenance by decoupling classification from service treatment.
- Enables policy reuse and modular design, reducing operational effort in large deployments.
- Allows flexible traffic behavior management, supporting multiple QoS features within a unified framework.
- Simplifies troubleshooting and verification by separating counters and policies per class and interface.
- Modular QoS command-line interface, on page 2

Modular QoS command-line interface

Modular QoS command-line interface (MQC) is a Quality of Service (QoS) CLI configuration framework that

- defines traffic classification rules using class-map objects
- associates classification with treatment actions for traffic packets in policy-map objects, and
- attaches traffic policies to interfaces (ingress or egress), thereby enabling consistent QoS control across platforms.

MQC enables users to create QoS policies by linking defined traffic classes with specific traffic treatment actions (marking, policing, shaping, queuing, or dropping) and then applying these policies to interfaces. The system enforces classification and subsequent treatment on packets flowing through the interfaces and enforces the QoS behavior described in the associated policy map.

MQC comprises of three fundamental logical components.

- Class-map: defines a traffic class used later by a policy-map as a match criteria to apply QoS treatment.
- Policy-map: associates one or more traffic classes with QoS actions such as marking, queueing, scheduling, shaping, WRED, or policing.
- Service-policy: applies a policy-map to an interface (ingress or egress), activating the configured QoS treatments for matching traffic.

MQC attributes

MQC exhibits several structural characteristics that define how it organizes and applies QoS policies.

- Implements a three-stage object model: classification (class-map), policy creation (policy-map), and policy attachment to interfaces (service-policy).
- Enables reuse of named objects (classes and policies) to reduce duplicated configuration.
- Distinguishes classification logic from traffic treatment logic so that traffic definition and QoS action remain decoupled.
- Includes default handling: traffic that doesn't match any class is bound to a default class with baseline behavior.
- Supports nested class-maps using match class-map to refine matching hierarchy.

Table 2: Differences between MQC and traditional QoS CLI frameworks

Attribute	MQC framework	Traditional QoS CLI framework
Type	Configuration framework consists of modular command hierarchy	Interface-centric command model consists of flat or monolithic CLI
Structure	Uses modular objects: class-map, policy-map, and service-policy	Defines QoS directly under each interface with feature-specific commands
Reusability	High—classes and policies are reusable across interfaces and services	Low—each interface must be configured independently
Consistency	Ensures identical QoS behavior across multiple interfaces	Risk of inconsistency due to repetitive manual configuration
Maintainability	Simplifies updates—change one policy and re-apply	Requires per-interface configuration changes for every modification

Benefits of MQC

MQC offers multiple operational advantages that enhance the consistency and scalability of QoS configuration.

- Promotes consistency in QoS configurations by using the same architecture on multiple platforms.
- Improves readability and maintenance by decoupling classification from service treatment.

- Enables policy reuse and modular design, reducing operational effort in large deployments.
- Allows flexible traffic behavior management, supporting multiple QoS features within a unified framework.
- Simplifies troubleshooting and verification by separating counters and policies per class and interface.