



Modular Quality of Service Command-Line Interface

This document describes the Modular Quality of Service Command-Line Interface (Modular QoS CLI) and contains the following sections:

- Overview, page 1
- Supported MIB, page 2
- Configuring QoS Policies Using the Modular QoS CLI, page 3
- Verifying the Configuration, page 7
- Modular QoS CLI Configuration Examples, page 7
- Command Reference, page 13

Overview

The Modular QoS CLI allows users to specify a traffic class independently of QoS policies.

The Modular QoS CLI is used to configure QoS. The Modular QoS CLI contains the following three steps, which are detailed more thoroughly later in this document:

-
- Step 1** Defining a traffic class with the **class-map** command
 - Step 2** Creating a service policy by associating the traffic class with one or more QoS policies (using the **policy-map** command)
 - Step 3** Attaching the service policy to the interface with the **service-policy** command
-

The **class-map** command is used to define a traffic class. A traffic class contains three major elements: a name, a series of **match** commands, and an instruction on how to evaluate these **match** commands. The traffic class is named in the **class-map** command line; for example, if you enter the **class-map don** command while configuring the traffic class in the command-line interface, the traffic class would be named don.

Match commands are used to specify various criteria for classifying packets. Packets are checked to see whether they match the criteria specified in the **match** commands; if a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS

specifications set in the service policy. Packets that fail to meet any of the matching criteria are classified as members of the default class. The default class is detailed more thoroughly in the “Configuring a Default Class” section of this document.

The instruction on how to evaluate these **match** commands is specified with one of the following two options: **class-map match-any** or **class-map match-all**. The functionality of these options is detailed more thoroughly in the “Using the class-map match-any and class-map match-all Commands” section of this document.

The **policy-map** command is used to associate a traffic class, which was defined by the **class-map** command, with one or more QoS policies. The result of this association is called a service policy. A service policy contains three elements: a name, a traffic class (specified with the **class** command), and the QoS policies (which are detailed in the “Creating a Service Policy” section of this document). The purpose of the service policy is to associate a traffic class with one or more QoS policies. The name of a service policy is specified in the **policy-map** command-line interface (for example, issuing the **policy-map gary** command would create a service policy named gary).

The Modular QoS CLI does not necessarily require that users associate only one traffic class to one service policy. When packets match to more than one match criterion, multiple traffic classes can be associated with a single service policy.

Similarly, the Modular QoS CLI allows multiple traffic classes (nested traffic classes, which are also called nested class maps) to be configured as a single traffic class. This can be achieved with the use of the **match class-map** command. The only method of combining match-any and match-all characteristics within a single traffic class is with the **match class-map** command. An example of a nested traffic class configuration is provided in the “Using a Traffic Class as a Match Criterion (Nested Class Maps)” section of this document.

The **service-policy** command is used to attach the service policy, as specified with the **policy-map** command, to an interface. Because the elements of the service policy can be applied to packets entering and leaving the interface, users are required to specify whether the service policy characteristics should be applied to incoming or outgoing packets. For instance, the **service-policy output gary** command would attach all the characteristics of the service policy named gary to the specified interface. All packets leaving the specified interface are evaluated according to the criteria specified in the service policy named Gary. For information on using the **service-policy** command, see the “Attaching a Service Policy to an Interface” section of this document.

Supported MIB

Class-Based Quality of Service Management Information Base

The Class-Based Quality of Service Management Information Base (Class-Based QoS MIB) provides read access to QoS configurations. This MIB also provides QoS statistics information based on the Modular QoS CLI, including information regarding class map and policy map parameters.

This Class-Based QoS MIB is actually two MIBs: CISCO-CLASS-BASED-QOS-MIB and CISCO-CLASS-BASED-QOS-CAPABILITY-MIB.

Use the Cisco Network Management Toolkit for MIBs tool on Cisco Connection Online (CCO) to locate MIBs.

Configuring QoS Policies Using the Modular QoS CLI

This section describes the tasks for configuring QoS functionality with the Modular QoS CLI. For a complete description of the commands mentioned in this section, refer to the *Quality of Service Solutions Command Reference*. The commands are listed alphabetically within the guide. To locate documentation of a specific command, use the command reference, master index, or online search.

Modular QoS CLI Configuration Task List

You must follow certain required, basic steps to configure the Modular QoS CLI. The following sections outline these tasks:

- Defining a Traffic Class, page 3
- Creating a Service Policy, page 5
- Attaching a Service Policy to an Interface, page 6

Defining a Traffic Class

The **class-map** command is used to create a traffic class. To create a traffic class containing match criteria, use the **class-map** command to specify the traffic class name, and then use a **match** command in class map configuration mode.

The syntax of the **class-map** command is:

```
class-map [match-any | match-all] class-name  
no class-map [match-any | match-all] class-name
```

The **class-map match-all** command is used when all of the match criteria in the traffic class must be met in order for a packet to match the specified traffic class. The **class-map match-any** command is used when the first possible match criterion from a list of match criteria must be met for a packet to match the specified traffic class. If neither match-all nor match-any is specified, the traffic class will behave in a manner consistent with **class-map match-all** command.

The **match not** command, rather than identifying the specific match parameter to use as a match criterion, is used to specify a match criterion that prevents a packet from being classified as a member of the class. For instance, if the **match not qos-group 6** command is issued while you configure the traffic class, QoS group 6 becomes the only QoS group value that is not considered a successful match criterion. All other QoS group values would be successful match criteria.

For additional information on using the match-any and match-all options, see the “Using the class-map match-any and class-map match-all Commands” section of this document.



Note

This document lists some of the command options in class map configuration mode. These command options vary among platforms and Cisco IOS releases. Because software is updated frequently, this list of commands might not represent the most updated software command options. For the most current command options for your Cisco IOS software, see the New Feature Documentation index for your particular Cisco IOS software release on CCO.

Command	Purpose
Router(config)# class-map <i>class-map-name</i>	Specifies the user-defined name of the traffic class.
Router(config)# class-map match-all <i>class-map-name</i>	Specifies a logical AND operator for all matching statements under this traffic class.
Router(config)# class-map match-any <i>class-map-name</i>	Specifies a logical OR operator for all matching statements under this traffic class.
Router(config-cmap)# match access-group <i>access-group</i>	Specifies the numbered access list against whose contents packets are checked to determine if they belong to the class.
Router (config-cmap)# match any	Specifies that all packets will be matched.
Router (config-cmap)# match class-map <i>class-name</i>	Specifies the name of a traffic class to be used as a matching criterion (for nesting traffic class [nested class maps] within one another).
Router(config-cmap)# match cos <i>cos-number</i>	Specifies the CoS value against whose contents packets are checked to determine if they belong to the class.
Router (config-cmap)# match destination-address <i>mac address</i>	Specifies the name of the destination MAC address used as a match criterion against which packets are checked to determine if they belong to the class.
Router(config-cmap)# match input-interface <i>interface-name</i>	Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.
Router (config-cmap)# match ip dscp <i>number</i>	Specifies up to eight differentiated services code point (DSCP) values used as match criteria. The value of each service code point is between 0 and 63.
Router (config-cmap)# match ip precedence <i>number</i>	Specifies up to eight IP precedence values used as match criteria.
Router(config-cmap)# match ip rtp <i>starting-port-number port-range</i>	Specifies the RTP port as the match criterion.
Router(config-cmap)# match mpls experimental <i>mpls-values</i>	Specifies the MPLS values to use as match criterion against which packets are checked to determine if they belong to the class
Router (config-cmap)# match not <i>match-criteria</i>	Specifies a match criterion value that prevents packets from being classified as members of a specified traffic class. All other values of that particular match criterion belong to the class.
Router(config-cmap)# match protocol <i>protocol</i>	Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.
Router(config-cmap)# match qos-group <i>qos-group-value</i>	Specifies the number of the QoS group index used as a match criterion against which packets are checked to determine if they belong to the class.
Router (config-cmap)# match source-address <i>mac address</i>	Specifies the name of the source MAC address used as a match criterion against which packets are checked to determine if they belong to the class.

Creating a Service Policy

To configure a service policy, use the **policy-map** command to specify the service policy name, and then use the following configuration commands to associate a traffic class, which was configured with the **class-map** command, with one or more QoS policies. The traffic class is associated with the service policy when the **class** command is used. The **class** command has to be issued immediately after entering policy map configuration mode. After entering the **class** command, you are automatically in policy map class configuration mode, which is where the QoS policies for the service policy are defined.

The QoS policies that can be applied in the service policy in policy map class configuration mode are detailed below.

The syntax of the **policy-map** command is:

```
policy-map policy-name
no policy-map policy-name
```

The syntax of the **class** command is:

```
class class-name
no class class-name
```

If a default class is configured, all traffic that fails to meet the matching criteria belongs to the default class.



Note

This document lists some of the command options for the policy map configuration mode. These command options vary among platforms and Cisco IOS releases. Because software is updated frequently, this list of commands might not represent the most updated software command options. For the most current command options for your Cisco IOS software, see the New Feature Documentation index for your particular Cisco IOS software release on CCO.

Command	Purpose
Router(config)# policy-map <i>policy-name</i>	Specifies the name of the service policy to configure.
Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class, which was defined with the class-map command, included in the service policy.
Router(config-pmap)# class class-default	Specifies the default class to be created as part of the service policy.
Router(config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	Specifies a minimum bandwidth guarantee to a traffic class. A minimum bandwidth guarantee can be specified in kilobits per second or by a percentage of the overall available bandwidth.
Router(config-pmap-c)# default <i>command</i>	Sets a command to its default value.
Router(config-pmap-c)# fair-queue <i>number-of-queues</i>	Specifies the number of queues to be reserved for the class.
Router(config-pmap-c)# police <i>bps burst-normal burst-max conform-action action exceed-action action violate-action action</i>	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm.

Command	Purpose
Router(config-pmap-c)# priority { <i>kpbs</i> percent <i>percent</i> } [<i>bytes</i>]	Specifies the guaranteed allowed bandwidth (in kbps or percentage) for priority traffic. The optional <i>bytes</i> argument controls the size of the burst allowed to pass through the system without being considered in excess of the configured kpbs rate.
Router(config-pmap-c)# queue-limit <i>packets</i>	Specifies the maximum number of packets queued for a traffic class (in the absence of the random-detect command).
Router(config-pmap-c)# random-detect	Enables a weighted random early detection (WRED) drop policy for a traffic class that has a bandwidth guarantee.
Router(config-pmap-c)# set atm-clp	Sets the ATM cell loss priority bit to 1.
Router(config-pmap-c)# set cos <i>cos-value</i>	Specifies a CoS value or values to associate with the packet. The number is in the range 0 to 7.
Router(config-pmap-c)# set ip dscp <i>ip-dscp-value</i>	Specifies the IP DSCP of packets within a traffic class. The IP DSCP value can be any value between 0 and 63.
Router(config-pmap-c)# set ip precedence <i>ip-precedence-value</i>	Specifies the IP precedence of packets within a traffic class. The IP precedence value can be any value between 0 and 7.
Router(config-pmap-c)# set mpls experimental <i>value</i>	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Router(config-pmap-c)# set qos-group <i>qos-group-value</i>	Specifies a QoS group value to associate with the packet. The QoS group value can be any value between 0 and 99.
Router(config-pmap-c)# service-policy <i>policy-map-name</i>	Specifies the name of a service policy to be used as a matching criterion (for nesting service policies [hierarchical service policies] within one another).
Router(config-pmap-c)# shape { average peak } <i>mean-rate</i> [<i>burst-size</i> [<i>excess-burst-size</i>]]	Shapes traffic to the indicated bit rate according to the algorithm specified.

Attaching a Service Policy to an Interface

Use the **service-policy** interface configuration command to attach a service policy to an interface and to specify the direction in which the policy should be applied (either on packets coming into the interface or packets leaving the interface).

Use the **no** form of the command to detach a service policy from an interface. The **service-policy** command syntax is:

```
service-policy {input | output} policy-map-name
no service-policy {input | output} policy-map-name
```



Note

Depending on the platform and Cisco IOS release, a service policy can be attached to an ATM permanent virtual circuit (PVC) subinterface, Frame Relay data-link connection identifier (DLCI), or other type of interface.

Command	Purpose
Router(config-if)# service-policy output <i>policy-map-name</i>	Specifies the name of the service policy to be attached to the output direction of the interface.
Router(config-if)# service-policy input <i>policy-map-name</i>	Specifies the name of the service policy to be attached to the input direction of the interface.

Verifying the Configuration

Use the **show class-map** *class-name* command to display the information relating to a traffic class. Use the **show policy-map** command to display the configuration of a service policy and its associated traffic classes. Forms of these commands are listed in the table below.

Command	Purpose
Router# show class-map	Displays all traffic class information.
Router# show class-map <i>class-name</i>	Displays the traffic class information for the user-specified traffic class.
Router# show policy-map	Displays all configured service policies.
Router# show policy-map <i>policy-map-name</i>	Displays the user-specified service policy.
Router# show policy-map <i>interface</i>	Displays configurations and statistics of all input and output policies, that are attached to an interface.
Router# show policy-map interface <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map interface <i>interface-spec</i> <i>input</i>	Displays configuration and statistics of the input policy attached to an interface.
Router# show policy-map interface <i>interface-spec</i> <i>output</i>	Displays configuration and statistics of the output policy attached to an interface.
Router# show policy-map [interface [<i>interface-spec</i> [<i>input</i> <i>output</i>] [class <i>class-name</i>]]]]	Displays the configuration and statistics of the class name configured in the policy.

Modular QoS CLI Configuration Examples

This section provides the following configuration examples:

- Defining Traffic Classes, page 8
- Creating a Service Policy, page 8
- Attaching a Service Policy to an Interface, page 8
- Using the match not Command, page 9
- Configuring a Default Class, page 9
- Using the class-map match-any and class-map match-all Commands, page 10

- Using a Traffic Class as a Match Criterion (Nested Class Maps), page 11
- Using a Service Policy as a QoS Policy (Hierarchical Service Policies), page 11

Defining Traffic Classes

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, access control list (ACL) 101 is used as the match criterion. For the second traffic class, called class2, ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit

Router(config)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

Creating a Service Policy

In the following example, a service policy called policy1 is defined to contain policy specifications for the two classes—class1 and class2. The match criteria for these classes were defined in the traffic classes (see the “Defining a Traffic Class” section).

For class1, the policy includes a bandwidth allocation request and a maximum packet count limit for the queue reserved for the class. For class2, the policy specifies only a bandwidth allocation request.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap)# exit

Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit
```

Attaching a Service Policy to an Interface

The following example shows how to attach an existing service policy (which was created in the preceding “Creating a Service Policy” section) to an interface. After you define a service policy with the **policy-map** command, you can attach it to one or more interfaces to specify the service policy for those interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached at the input and only one service policy attached at the output.

```
Router(config)# interface e1/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit

Router(config)# interface fa1/0/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```


Using the match not Command

The **match not** command is used to specify a specific QoS policy value that is not used as a match criterion. When using the **match not** command, all other values of that QoS policy become successful match criteria.

For instance, if the **match not qos-group 4** command is issued in class map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

In the following traffic class, all protocols except IP are considered successful match criteria:

```
Router(config)# class-map noip
Router(config-cmap)# match not protocol ip
Router(config-cmap)# exit
```

Configuring a Default Class

Unclassified traffic (traffic that does not meet the match criteria specified in the traffic classes) is treated as belonging to the default class.

If the user does not configure a default class, packets are still treated as members of the default class. However, by default, the default class has no enabled features. Therefore, packets belonging to a default class with no configured features have no QoS functionality. These packets are then placed into a FIFO (first-in, first-out) queue and forwarded at a rate determined by the available underlying link bandwidth. This FIFO queue is managed by tail drop. (Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full).

The following example configures a service policy for the default class of the service policy called policy1. The default class (which is always called class-default) has these characteristics: 10 queues for traffic that does not meet the match criteria of other classes whose policy is defined by the service policy policy1, and a maximum of 20 packets per queue before tail drop is enacted to handle additional enqueued packets.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue 10
Router(config-pmap-c)# queue-limit 20
```

Using the class-map match-any and class-map match-all Commands

This section illustrates the difference between the **class-map match-any** command and the **class-map match-all** command. The match-any and match-all options determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (match-all) or one of the match criteria (match-any) in order to be considered a member of the traffic class.

The following example shows a traffic class configured with the **class-map match-all** command:

```
Router(config)# class-map match-all john
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

If a packet arrives on a router with traffic class john configured on the interface, the packet is evaluated to see if it matches the IP protocol, QoS group 4, *and* access group 101. If all three of these match criteria are met, the packet matches traffic class john.

The following example shows a traffic class configured with the **class-map match-any** command:

```
Router(config)# class-map match-any george
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

In traffic class george, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to see whether IP protocol can be used as a match criterion. If IP protocol can be used as a match criterion, the packet is matched to traffic class george. If IP protocol is not a successful match criterion, then QoS group 4 is evaluated as a match criterion. Each matching criterion is evaluated to see if the packet matches that criterion. Once a successful match occurs, the packet is classified as a member of traffic class george. If the packet matches none of the specified criteria, the packet is classified as a member of the default class.

Note that the **class-map match-all** command requires that all of the match criteria must be met in order for the packet to be considered a member of the specified traffic class. However, only one match criterion must be met for the packet in the **class-map match-any** command to be classified as a member of the traffic class.

Using a Traffic Class as a Match Criterion (Nested Class Maps)

Traffic classes can be nested within one another, saving users the overhead of re-creating a new traffic class when most of the information exists in a previously configured traffic class. In the following example, traffic class `brando` has the same characteristics as traffic class `marlon`, with the exception that traffic class `brando` has added a destination address as a match criterion. Rather than configuring traffic class `brando` line by line, a user can enter the **match class-map marlon** command. This command allows all of the characteristics in traffic class `marlon` to be included in traffic class `brando`, and the user can simply add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any marlon
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# class-map match-all brando
Router(config-cmap)# match class-map marlon
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# exit
```

The only method of including both `match-any` and `match-all` characteristics in a single traffic class is to use the **match class-map** command. To combine `match-any` and `match-all` characteristics into a single class, a traffic class created with the `match-any` instruction must use a class configured with the `match-all` instruction as a match criterion (through the **match class-map** command), or vice versa.

The following example shows how to combine the characteristics of two traffic classes, one with `match-any` and one with `match-all` characteristics, into one traffic class with the **match class-map** command. The end result of traffic class `montana` requires a packet to match one of the following three match criteria to be considered a member of traffic class `montana`: IP protocol *and* QoS group 4, destination MAC address 1.1.1, or access group 2.

In this example, only traffic class `montana` is used with service policy `niners`.

```
Router(config)# class-map match-all joe
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit

Router(config)# class-map match-any montana
Router(config-cmap)# match class-map joe
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# policy-map niners
Router(config-pmap)# class montana
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# exit
```

Using a Service Policy as a QoS Policy (Hierarchical Service Policies)

A service policy can be nested within a QoS policy when the **service-policy** command is used in policy map class configuration mode. A service policy that contains a nested service policy is called a hierarchical service policy.

A hierarchical service policy contains a child and a parent policy. The child policy is the previously defined service policy that is being associated with the new service policy through the use of the **service-policy** command. The new service policy using the preexisting service policy is the parent policy. In the example in this section, service policy child is the child policy and service policy parent is the parent policy.

Hierarchical service policies can be attached to subinterfaces, Frame Relay PVCs, and ATM PVCs. A hierarchical service policy is particularly beneficial when configuring Versatile Interface Processor-Based Distributed FRF.11 and FRF.12 PVCs. By using hierarchical service policies, a single service policy (with a child and a parent policy) can be used to shape and prioritize PVC traffic. In the following example, the child policy is responsible for prioritizing traffic while the parent policy is responsible for shaping traffic. In this configuration, the parent policy allows packets to be sent from the interface, and the child policy determines the order in which the packets are sent.

```
Router(config)# policy-map child
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50

Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# service-policy child
```

With the exception that the values associated with the **priority** and **shape** commands can be modified, the above example is the required configuration for PVCs using FRF.11 or FRF.12. The value used with the **shape** command is provisioned from the cir value from the service provider. For additional information on FRF.11 and FRF.12 PVCs, see the *Versatile Interface Processor-Based Distributed FRF.11 and FRF.12* document on CCO.

For additional information on hierarchical service policies, see the “Creating a Service Policy for FRF.11 and FRF.12” section and the **service-policy** command reference section of *Versatile Interface Processor-Based Distributed FRF.11 and FRF.12* document on CCO.

Command Reference

This section documents commands associated with the Modular QoS CLI. All other commands used with this feature are documented in the Cisco IOS Release 12.1 command reference publications.



Note This section does not document commands for the QoS policies that can be used in policy map class configuration mode. For the command references for these QoS policies, see the individual feature modules associated with these features on CCO or the Documentation CD-ROM.

This section contains the following command references:

- **class**
- **class-map**
- **match access-group**
- **match any**
- **match class-map**
- **match cos**
- **match destination-address mac**
- **match input-interface**
- **match ip dscp**
- **match ip precedence**
- **match ip rtp**
- **match not**
- **match protocol**
- **match qos-group**
- **match source-address mac**
- **policy-map**
- **service-policy**
- **show policy**
- **show policy-map class**
- **show policy-map interface**

class

To specify the name of the class whose policy you want to create or change or to specify the default class called **class-default** before you configure its policy, use the **class** policy map configuration command. To remove a class from the policy map, use the **no** form of this command.

class [*class-name* | **class-default**]

no class [*class-name* | **class-default**]

Syntax Description	<i>class-name</i>	The name of the class for which you want to configure or modify policy.
	class-default	Specifies the default class so that you can configure or modify its policy.

Defaults No default behavior.

Command Modes Policy map configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines Enter the **policy-map** command to identify the policy map and enter policy map configuration mode before you use the **class** command. After you specify a policy map, you can configure policy for new classes or modify policy for any existing classes in that policy map.

The class name that you specify in the policy map ties the characteristics for that class—that is, its policy—to the class map and its match criteria configured using the **class-map** command.

When you configure policy for a class and specify its bandwidth and attach the policy map to an interface, class-based weighted fair queuing (CBWFQ) checks to determine if the bandwidth requirement of the class can be satisfied. If so, CBWFQ allocates a queue for the bandwidth requirement.

When a class is removed, available bandwidth for the interface is incremented by the amount previously allocated to the class.

The maximum number of classes you can configure for a router—and, therefore, within a policy map—is 64.

Examples

The following example configures policy for a class named `acl136` included in the policy map called `policy1`. Class `acl136` has these characteristics: a minimum of 2000 kilobits per second (kbps) of bandwidth are expected to be delivered to this class in the event of congestion, and the queue reserved for this class can enqueue 40 packets before tail drop is enacted. Note that when the policy map containing this class is attached to the interface to stipulate the service policy for that interface, available bandwidth is assessed, taking into account all class policies and RSVP, if configured.

```
policy-map policy1
class acl136
bandwidth 2000
queue-limit 40
```

The following example configures policy for a class named `int101` included in the policy map called `policy8`. Class `int101` has these characteristics: a minimum of 3000 kbps of bandwidth are expected to be delivered to this class in the event of congestion, and a weight factor of 10 is used to calculate the average queue size. For congestion avoidance, WRED packet drop is used, not tail drop. Note that when the policy map containing this class is attached to the interface to stipulate the service policy for that interface, available bandwidth is assessed.

```
policy-map policy8
class int101
bandwidth 3000
random-detect exponential-weighting-constant 10
```

The following example configures policy for the **class-default** default class included in the policy map called `policy1`. The **class-default** default class has these characteristics: 10 hashed queues for traffic that does not meet the match criteria of other classes whose policy is defined by the policy map `policy1`, and a maximum of 20 packets per queue before tail drop is enacted to handle additional enqueued packets.

```
policy-map policy1
class class-default
fair-queue 10
queue-limit 20
```

The following example configures policy for the **class-default** default class included in the policy map called `policy8`. The **class-default** default class has these characteristics: 20 hashed queues for traffic that does not meet the match criteria of other classes whose policy is defined by the policy map `policy8` and a weight factor of 14 is used to calculate the average queue size. For congestion avoidance, WRED packet drop is used, not tail drop.

```
policy-map policy8
class class-default
fair-queue 20
random-detect exponential-weighting-constant 14
```

Related Commands

Command	Description
bandwidth	Specifies or modifies the bandwidth allocated for a class whose policy is specified in the policy map.
policy-map	Specifies the policy map to which the class policy belongs.
queue-limit	Specifies or modifies the maximum number of packets that can accumulate in the queue reserved for the class before tail drop or (if WRED is configured as part of the class policy) packet drop is enacted.

random-detect exponential-weighting-constant Configures the exponential weight factor used in calculating the average queue length.

random-detect precedence Configures the parameters for packets with a specific IP Precedence.

class-map

To create a class map to be used for matching packets to the class whose name you specify, use the **class-map** global configuration command. To remove an existing class map from the router, use the **no** form of this command.

class-map *class-map-name*

no class-map *class-map-name*

Syntax Description	<i>class-map-name</i>	Name of the class for the class map. The class name is used for both the class map and to configure policy for the class in the policy map.
---------------------------	-----------------------	---

Defaults No default behavior.

Command Modes Global configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines Use this command to specify the name of the class for which you want to create or modify class map match criteria. Use of the **class-map** command enables class map configuration mode in which you can enter one of the match commands to configure the match criteria for this class. Packets arriving at the output interface are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

Examples The following example specifies access101 as the name of a class, and it defines a class map for this class. The **match access-group** command is entered following the **class-map** command to configure the numbered access control list (ACL) 101, whose contents are to be used as match criteria for the class.

```
class-map access101
  match access-group 101
```

Related Commands	Command	Description
	class	Specifies a class belonging to a service policy map.

class class-default	Specifies the default class for a service policy map.
match access-group	Configures the match criteria for a class map based on the specified ACL number.
match input-interface	Configures the match criteria for a class map based on the specified input interface.
match protocol	Configures the match criteria for a class map based on the specified protocol.
policy-map	Specifies the policy map to which the class belongs.

match access-group

To configure the match criteria for a class map based on the specified access-control list (ACL) number, use the **match access-group** class map configuration command. To remove ACL match criteria from a class map, use the **no** form of this command.

match access-group *access-group*

no match access-group *access-group*

Syntax Description	<i>access-group</i>	A numbered ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class.
---------------------------	---------------------	---

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines	Traffic classes are defined based on match criteria including numbered ACLs, protocols, and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. The match access-group command specifies a numbered ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map. Use the class-map command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use the match access-group command to configure the match criteria.
-------------------------	---

Examples	The following example specifies a class map called acl144 and configures the ACL numbered 144 to be used as the match criteria for this class.
-----------------	--

```
class-map acl144
  match access-group 144
```

Related Commands	Command	Description
	class-map	Creates a class map for the specified class.
	match input-interface	Configures the match criteria for a class map based on the specified input interface.
	match protocol	Configures the match criteria for a class map based on the specified protocol.

match any

To configure the match criteria for a class map to be successful match criteria for all packets, use the **match any** class map configuration command. To remove all criteria as successful match criteria, use the **no** form of this command.

match any

no match any

Syntax Description

None

Defaults

No default behavior.

Command Modes

Class map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced in Release 12.0 XE.
12.0(5)T	This command was introduced in Release 12.0 T.
12.1(1)E	This command was introduced in Release 12.1 E.

Examples

In the following configuration, all packets leaving interface e1/1 will be policed based on the parameters specified in policy map class configuration mode.

```
Router(config)# class-map matchany
Router(config-cmap)# match any
Router(config-cmap)# exit

Router(config)# policy-map niners
Router(config-pmap)# class montana
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# exit

Router(config)# interface e1/1
Router(config-if)# service-policy output policy1
```

■ match any

Related Commands	Command	Description
	class-map	Creates a class map for the specified class.
	match input-interface	Configures the match criteria for a class map based on the specified input interface.
	match protocol	Configures the match criteria for a class map based on the specified protocol.

match class-map

To use a traffic class as a classification policy, use the **match class-map** class map configuration command. To remove a specific traffic class as a match criterion, use the **no** form of this command.

match class-map *class-map-name*

no match class-map *class-map-name*

Syntax Description	<i>class-map-name</i>	(Required) Specifies the name of the traffic class to use as a match criterion.
---------------------------	-----------------------	---

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)XE	This command was first introduced.
	12.1(1)E	This command was introduced in Release 12.1(1)E.
	12.1(5)T	This command was integrated into Release 12.1 T.

Usage Guidelines	<p>The only method of including both match-any and match-all characteristics in a single traffic class is to use the match class-map command. To combine match-any and match-all characteristics into a single class, a traffic class created with the match-any instruction must use a class configured with the match-all instruction as a match criterion (through the match class-map command), or vice versa.</p> <p>By using the match class-map command, traffic classes can be nested within one another, saving users the overhead of re-creating a new traffic class when most of the information exists in a previously configured traffic class.</p>
-------------------------	---

Examples	<p>In the following example, traffic class brando has the same characteristics as traffic class marlon, with the exception that traffic class brando has added a destination address as a match criterion. Rather than configuring traffic class brando line by line, a user can enter the match class-map marlon command.</p>
-----------------	---

This command allows all of the characteristics in traffic class marlon to be included in traffic class brando, and the user can simply add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any marlon
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
```

```
Router(config)# class-map match-all brando
Router(config-cmap)# match class-map marlon
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# exit
```

The following example shows how to combine the characteristics of two traffic classes, one with match-any and one with match-all characteristics, into one traffic class with the **match class-map** command. The end result of traffic class montana requires a packet to match one of the following three match criteria to be considered a member of traffic class montana: IP protocol *and* QoS group 4, destination MAC address 1.1.1, or access group 2.

In this example, only traffic class montana is used with service policy niners.

```
Router(config)# class-map match-all joe
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit
```

```
Router(config)# class-map match-any montana
Router(config-cmap)# match class-map joe
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
```

```
Router(config)# policy-map niners
Router(config-pmap)# class montana
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# exit
```

Related Commands

Command	Description
class-map	Creates a class map for the specified class.

match cos

To match a packet based on a Layer 2 Class of Service marking, use the **match cos** command in class map configuration mode. To remove a specific Layer 2 Class of Service/ISL marking, use the **no** form of this command:

```
match cos cos-value [cos-value cos-value cos-value]
```

```
no match cos cos-value [cos-value cos-value cos-value]
```

Syntax Description	cos	Specifies matching of IEEE 802.1Q/ISL Class of Service values.
	<i>cos-value</i>	Specific IEEE 802.1Q/ISL Class of Service value. The <i>cos-value</i> is between 0 and 7, and up to four CoS values can be specified in one match cos statement.
Defaults	Disabled	
Command Modes	Class map configuration	
Command History	Release	Modification
	12.1(5)T	This command was introduced.

Examples

In the following example, the cos-values of 1, 2, and 3 are successful match criteria for the interface containing the classification policy cos:

```
Router(config)# class-map cos
Router(config-cmap)# match cos 1 2 3
```

In the following example, classes voice and video-n-data are created to classify traffic based on the CoS values. QoS treatment is then given to the appropriate packets (in this case, the QoS treatment is priority 64 and bandwidth 512) in the cos-based-treatment policy map.

```
Router(config)# class-map voice
Router(config-cmap)# match cos 7

Router(config)# class-map video-n-data
Router(config-cmap)# match cos 5

Router(config)# policy-map cos-based-treatment
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 64
Router(config-pmap-c)# exit
Router(config-pmap)# class video-n-data
Router(config-pmap-c)# bandwidth 512
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# interface fa0/0.1
Router(config-if)# service-policy output cos-based-treatment
```

The service policy configured in this section is attached to all packets leaving fastethernet interface 0/0.1. The service policy can be attached to any interface that supports service policies.

Related Commands	Command	Description
	class-map	Specifies the user-defined name of the traffic class.
	policy-map	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
	service-policy	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
	set cos	Marks the Layer 2 CoS value of an outgoing packet.
	show class-map	Displays configuration information of all class maps.

match destination-address mac

To use the destination MAC address as a match criterion in class map configuration mode, use the **match destination-address mac** class map configuration command. To remove a previously specified destination MAC address as a match criterion in class map configuration mode, use the **no match destination-address mac** command.

match destination-address mac *address*

no match destination-address mac *address*

Syntax Description	<i>address</i>	(Required) Specifies the specific destination MAC address to be used as a match criterion.
---------------------------	----------------	--

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)XE	This command was first introduced.
	12.1(1)E	This command was introduced for Release 12.1 E.
	12.1(5)T	This command was integrated into Release 12.1 T.

Examples The following example specifies a class map called macaddress and specifies the destination mac address to be used as the match criteria for this class.

```
class-map macaddress
match destination-address mac 00:00:00:00:00:00
```

Related Commands	Command	Description
	class-map	Creates a class map for the specified class.

match input-interface

To configure a class map to use the specified input interface as match criterion, use the **match input-interface** class map configuration command. To remove the input interface match criterion from a class map, use the **no** form of this command.

match input-interface *interface-name*

no match input-interface *interface-name*

Syntax Description	<i>interface-name</i>	Name of the input interface to be used as match criteria.
---------------------------	-----------------------	---

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines	The match input-interface command specifies the name of an input interface to be used as the match criterion against which packets are checked to determine if they belong to the class specified by the class map. Use the class-map command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use the match input-interface command to configure its match criteria.
-------------------------	--

Examples	The following example specifies a class map called eth1 and configures the input interface named ethernet1 to be used as the match criterion for this class:
-----------------	--

```
class-map eth1
  match input-interface ethernet1
```

Related Commands

Command	Description
class-map	Creates a class map for the specified class.
match access-group	Configures the match criteria for a class map based on the specified numbered ACL.
match protocol	Configures the match criteria for a class map based on the specified protocol.

match ip dscp

To identify a specific IP differentiated service code point (DSCP) value as a match criterion, use the **match ip dscp** command in class map configuration mode. To remove a specific IP DSCP value from a class map, use the **no** form of this command.

```
match ip dscp ip-dscp-value [ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value
ip-dscp-value ip-dscp-value ip-dscp-value]
```

```
no match ip dscp ip-dscp-value [ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value
ip-dscp-value ip-dscp-value ip-dscp-value]
```

Syntax Description

ip	Specifies IP-specific packet matching.
dscp	Specifies IP dscp value matching.
<i>ip-dscp-value</i>	Specifies the exact value between 0 and 63 used to identify an IP DSCP value.

Defaults

No default behavior or values.

Command Modes

Class map configuration

Command History

Release	Modification
12.0(5)XE	This command was first introduced.
12.0(9)S	This command was introduced in Release 12.0 S.
12.1(2)T	This command was introduced in Release 12.1 T

Usage Guidelines

Up to eight IP DSCP values can be matched in one match statement. For example, if you wanted the IP DSCP values of 0, 1, 2, 3, 4, 5, 6, 7 (note that only one of the IP DSCP values has to be a successful match criterion, not all of the specified IP DSCP values), enter the following command:

```
match ip dscp 0 1 2 3 4 5 6 7
```

This command is used by the class map to identify a specific IP DSCP value marking on a packet. The *ip-dscp-values* are used as markings only. The IP DSCP values have no mathematical significance. For instance, the *ip-dscp-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *ip-dscp-value* of 2 is different than a packet marked with the *ip-dscp-value* of 1. The treatment of these marked packets is defined by the user through the setting of QoS policies in policy map class configuration mode.

Examples

The following example shows how to configure service policy priority50 and attach service policy priority50 to an interface. In this example, class map ipdscp15 will evaluate all packets entering interface fast ethernet 1/0/0 for an ip DSCP value of 15. If the incoming packet has been marked with the IP DSCP value of 15, the packet will be treated with a priority level of 55.

```
Router(config)# class-map ipdscp15
Router(config-cmap)# match ip dscp 15
Router(config)# #exit
Router(config)# #policy-map priority55
Router(config-pmap)# #class ipdscp15
Router(config-pmap-c)# priority 55
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy input priority55
```

Related Commands

Command	Description
set ip dscp	Marks the IP DSCP value for packets within a traffic class. The IP DSCP value can be any value between 0 and 63.
class-map <i>class-name</i>	Specifies the user-defined name of the traffic class.
policy-map <i>policy-name</i>	Specifies the name of the service policy to configure.
service-policy <i>policy-map-name</i>	Specifies the name of the service policy to be attached to the interface.
show class-map	Displays all configured traffic classes.
show class-map <i>class-map-name</i>	Displays the user-specified class map.

match ip precedence

To identify IP precedence values as match criteria, use the **match ip precedence** command in class map configuration mode. To remove IP precedence values from a class map, use the **no** form of this command.

match ip precedence *ip-precedence-value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]

no match ip precedence *ip-precedence value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]

Syntax Description	ip	Specifies IP-specific packet matching.
	precedence	Specifies IP precedence value matching.
	<i>ip-precedence-value</i>	Specifies the exact value between 0 and 7 used to identify an IP precedence value.

Defaults No default behavior or values.

Command Modes Class map configuration

Command History	Release	Modification
	12.0(5)XE	This command was first introduced.
	12.1(5)T	This command was integrated into Release 12.1 T.

Usage Guidelines Up to four precedence values can be matched in one match statement. For example, if you wanted the IP precedence values of 0, 1, 2, or 3 (note that only one of the IP precedence values has to be a successful match criterion, not all of the specified IP precedence values), enter the following command:

match ip precedence 0 1 2 3

The *ip-precedence-values* are used as markings only. The IP precedence values have no mathematical significance. For instance, the *ip-precedence-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *ip-precedence-value* of 2 is different than a packet marked with the *ip-precedence-value* of 1. The treatment of these different packets is defined by the user through the setting of QoS policies in policy map class configuration mode.

Examples

The following example shows how to configure service policy priority50 and attach service policy priority50 to an interface. In this example, class map ipprec5 will evaluate all packets entering interface fast ethernet 1/0/0 for an IP precedence value of 5. If the incoming packet has been marked with the IP precedence value of 5, the packet will be treated with a priority level of 50.

```
Router(config)# class-map ipprec5
Router(config-cmap)# match ip precedence 5
Router(config)# #exit
Router(config)# #policy-map priority50
Router(config-pmap)# #class ipprec5
Router(config-pmap-c)# priority 50
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy input priority50
```

Related Commands

Command	Description
set ip precedence	Specifies an IP precedence value for packets within a traffic class. The IP precedence value can be any value between 0 and 7.
class-map <i>class-map-name</i>	Specifies the user-defined name of the traffic class.
policy-map <i>policy-name</i>	Specifies the name of the service policy to configure.
service-policy <i>policy-map-name</i>	Specifies the name of the service policy to be attached to the interface.
show class-map	Displays all configured traffic classes.
show class-map <i>class-map-name</i>	Displays the user-specified class map.

match ip rtp

To configure a class map to use the Real-Time Protocol (RTP) protocol port as the match criteria, use the **match ip rtp** class-map configuration command. To remove the RTP protocol port match criteria, use the **no** form of this command.

match ip rtp *starting-port-number port-range*

no match ip rtp

Syntax Description

<i>starting-port-number</i>	The starting RTP port number. Values range from 2000 to 65535.
<i>port-range</i>	The RTP port number range. Values range from 0 to 16383.

Defaults

No default behavior.

Command Modes

Class map configuration

Command History

Release	Modification
12.1(2)T	This command was first introduced.

Usage Guidelines

This command is used to match IP RTP packets that fall within the specified port range. It matches packets destined to all even UDP port numbers in the range <starting port range> <starting port range + port range>.

Use of an RTP port range as the match criterion is particularly effective for applications that use RTP, such as voice or video.

Examples

The following example specifies a class map called eth1 and configures the RTP port number 2024 and range 1000 to be used as the match criterion for this class:

```
class-map eth1
 match ip rtp 2024 1000
```

Related Commands

Command	Description
ip rtp priority	Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.
match access-group	Configures the match criteria for a class map based on the specified ACL.

match mpls experimental

To configure a class map to use the specified value of the EXP field as a match criterion, use the **match mpls experimental** class map configuration command. To remove the EXP field match criterion from a class map, use the **no** form of this command.

match mpls experimental *number*

no match mpls experimental *number*

Syntax Description	<i>number</i>	The EXP field value to be used as match criteria. Any number from 0 to 7.
---------------------------	---------------	---

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(7)XE1	This command was first introduced.
12.1(1)E	This command was integrated into Release 12.1 E.	
12.1(5)T	This command was integrated into Release 12.1 T.	

Examples	The following example specifies a class map called eth1 and configures the MPLS values of 1 and 2 to be used as the match criterion for this class:
-----------------	---

```
class-map eth1
  match mpls experimental 1 2
```

match not

To specify the single match criterion value to use as an unsuccessful match criterion in class map configuration mode, use the **match not** class map configuration command. To remove a previously specified source value to not use as a match criterion, use the **no match not** command.

match not *match-criteria*

no match not *match-criteria*

Syntax Description	<i>match-criteria</i>	(Required) Specifies the match criterion value that is an unsuccessful match criterion. All other values of the specified match criterion will be considered successful match criteria.
---------------------------	-----------------------	---

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)XE	This command was first introduced.
	12.0(5)T	This command was integrated into Cisco IOS Release 12.1 T.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines	The match not command is used to specify a specific QoS policy value that is not used as a match criterion. When using the match not command, all other values of that QoS policy become successful match criteria.
-------------------------	---

For instance, if the **match not qos-group 4** command is issued in class map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

Examples	In the following traffic class, all protocols except IP are considered successful match criteria:
-----------------	---

```
Router(config)# class-map noip
Router(config-cmap)# match not protocol ip
Router(config-cmap)# exit
```

Related Commands	Command	Description
	class-map	Creates a class map for the specified class.

match protocol

To configure the match criteria for a class map based on the specified protocol, use the **match protocol** class map configuration command. To remove protocol-based match criteria from a class map, use the **no** form of this command.

match protocol *protocol*

no match protocol *protocol*

Syntax Description	<i>protocol</i>	Name of the protocol to match against. The following protocols are supported:
		aarp —AppleTalk ARP
		apollo —Apollo Domain
		arp —IP ARP
		bridge —Bridging
		bstun —Block Serial Tunnel
		cdp —Cisco Discovery Tunnel
		clns —ISO CLNS
		clns_es —ISO CLNS End System
		clns_is —ISO CLNS Intermediate System
		cmns —ISO CMNS
		compressedtcp —Compressed TCP
		decnet —DECnet
		decnet_node —DECnet Node
		decnet_router-I1 —DECnet Router L1
		decnet_router-I2 —DECnet Router L2
		dls —Data Link Switching
		ip —IP
		ipx —Novell IPX
		llc2 —llc2
		pad —PAD links
		qllc —qllc protocol
		rsrb —Remote Source-Route Bridging
		snapshot —Snapshot routing support
		stun —Serial Tunnel
		vines —Banyan Vines
		xns —Xerox Network Services

Defaults No default behavior.

Command Modes Class map configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines The **match protocol** command specifies the name of a protocol to be used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map. Use the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use the **match protocol** command to configure the match criteria for it.

Examples The following example specifies a class map called ipx and configures the ipx protocol as match criteria for it:

```
class-map ipx
match protocol ipx
```

match qos-group

To identify a specific QoS group value as a match criterion, use the **match qos-group** command in class map configuration mode. To remove a specific QoS group value from a class map, use the **no** form of this command.

match qos-group *qos-group-value*

no match qos-group *qos-group-value*

Syntax Description	qos-group	Specifies QoS group value matching.
	<i>qos-group-value</i>	Specifies the exact value between 0 and 99 used to identify a QoS group value.

Defaults No default behavior or values.

Command Modes Class map configuration

Command History	Release	Modification
	12.0(5)XE	This command was introduced.
	12.0(9)S	This command was introduced on the 12.0 S train.

Usage Guidelines This command is used by the class map to identify a specific QoS group value marking on a packet. The *qos-group-values* are used as markings only. The QoS group values have no mathematical significance. For instance, the *qos-group-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *qos-group-value* of 2 is different than a packet marked with the *qos-group-value* of 1. The treatment of these packets is defined by the user through the setting of QoS policies in policy map class configuration mode.

The QoS group value is local to the router, meaning that the QoS group value that is marked on a packet does not leave the router when the packet leaves the router. If you need a marking that resides in the packet, use IP precedence setting, IP DSCP setting, or another method of packet marking.

Examples

The following example shows how to configure service policy priority50 and attach service policy priority50 to an interface. In this example, class map qosgroup5 will evaluate all packets entering interface fast ethernet 1/0/0 for a QoS group value of 5. If the incoming packet has been marked with the QoS group value of 5, the packet will be treated with a priority level of 50.

```
Router(config)# class-map qosgroup5
Router(config-cmap)# match qos-group 5
Router(config)# #exit
Router(config)# #policy-map priority50
Router(config-pmap)# #class qosgroup5
Router(config-pmap-c)# priority 50
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy output priority50
```

Related Commands

Command	Description
set ip precedence	Specifies an IP precedence value for packets within a traffic class. The IP precedence value can be any value between 0 and 7.
class-map <i>class-name</i>	Specifies the user-defined name of the traffic class.
policy-map <i>policy-name</i>	Specifies the name of the service policy to configure.
service-policy <i>policy-map-name</i>	Specifies the name of the service policy to be attached to the interface.
show class-map	Displays all configured traffic classes.
show class-map <i>class-map-name</i>	Displays the user-specified class map.

match source-address mac

To use the source MAC address as a match criterion in class map configuration mode, use the **match source-address mac** class map configuration command. To remove a previously specified source MAC address as a match criterion in class map configuration mode, use the **no match source-address mac** command.

match source-address mac *address*

no match source-address mac *address*

Syntax Description	<i>address</i>	(Required) Specifies the source destination MAC address to be used as a match criterion.
---------------------------	----------------	--

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Class map configuration
----------------------	-------------------------

Command History	Release	Modification
	12.0(5)XE	This command was first introduced.
12.1(1)E	This command was introduced in Release 12.1 E.	
12.1(5)T	This command was integrated into Release 12.1 T.	

Usage Guidelines	<p>This command can only be used on an input interface with a MAC address. These interfaces include FastEthernet and Ethernet interfaces.</p> <p>This command cannot be used on output interfaces with no MAC address, such as serial and ATM interfaces.</p>
-------------------------	---

Examples	<pre>class-map matchsrcmac match source-address mac 0.0.0</pre>
-----------------	---

Related Commands	Command	Description
	class-map	Creates a class map for the specified class.

policy-map

To create or modify a policy map that can be attached to one or more interfaces to specify a service policy, use the **policy-map** global configuration command. To delete a policy map, use the **no** form of this command.

policy-map *policy-map-name*

no policy-map *policy-map-name*

Syntax Description	<i>policy-map-name</i>	Name of the policy map.
---------------------------	------------------------	-------------------------

Defaults	No default behavior or values.
-----------------	--------------------------------

Command Modes	Global configuration
----------------------	----------------------

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines Use the **policy-map** command to specify the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map. Entering the **policy-map** command enables policy map configuration mode in which you can configure or modify the class policies for that policy map.

You can configure class policies in a policy map only if the classes have match criteria defined for them. You use the **class-map** and **match** commands to configure the match criteria for a class. Because you can configure a maximum of 64 class maps, no policy map can contain more than 64 class policies.

A single policy map can be attached to multiple interfaces concurrently. When you attempt to attach a policy map to an interface, the attempt is denied if the interface's available bandwidth cannot accommodate the total bandwidth requested by class policies comprising the policy map. In this case, if the policy map is already attached to other interfaces, it is removed from them.

Whenever you modify class policy in an attached policy map, CBWFQ is notified and the new classes are installed as part of the policy map in the CBWFQ system.

Examples

The following example creates a policy map called policy9 and configures three class policies to belong to that map. Of these classes, two specify policy for classes with class maps that specify match criteria based on either a numbered access control list (ACL) or an interface name, and one specifies policy for the default class called **class-default** to which packets that do not satisfy configured match criteria are directed.

```

policy-map policy9

class acl136
  bandwidth 2000
  queue-limit 40

class ethernet101
  bandwidth 3000
  random-detect exponential-weighting-constant 10

class class-default
  fair-queue 10
  queue-limit 20

```

Related Commands

Command	Description
bandwidth	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
class	Specifies the class whose bandwidth is to be configured or modified.
class class-default	Specifies the default class whose bandwidth is to be configured or modified.
queue-limit	Specifies or modifies the maximum number of packets that can accumulate in the queue reserved for a class of a policy map.
random-detect exponential-weighting-constant	Configures the exponential weight factor used in calculating the average queue length.
random-detect precedence	Configures the parameters for packets with a specific IP Precedence. The minimum threshold for IP Precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence.
service-policy output	Enables CBWFQ and attaches the specified service policy map to the interface.

service-policy

To attach a policy map to an input interface or an output interface to be used as the service policy for that interface, use the **service-policy** global command. To remove a service policy from an input or output interface, use the **no** form of this command.

service-policy {**input** | **output**} *policy-map-name*

no service-policy {**input** | **output**} *policy-map-name*

Syntax Description	input	Attaches the specified policy map to the input interface.
	output	Attaches the specified policy map to the output interface.
	<i>policy-map</i>	The name of a service policy map (created using the policy-map command) to be attached.

Defaults No default behavior

Command Modes Global configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines You can attach a single policy map to one or more interfaces to specify the service policy for those interfaces.

To successfully attach a policy map to an interface, the aggregate of the configured minimum bandwidths of the classes comprising the policy map must be less than or equal to 75 percent of the interface bandwidth.

Attaching a service policy and enabling CBWFQ on an interface renders ineffective any commands related to fancy queueing such as commands pertaining to fair queueing, custom queueing, priority queueing, and Weighted Random Early Detection (WRED). You can configure these features only after you remove the policy map from the interface.

You can modify a policy map attached to an interface, changing the bandwidth of any of the classes comprising the map. Bandwidth changes that you make to an attached policy map are effective only if the aggregate of the bandwidth amounts for all classes comprising the policy map, including the modified class bandwidth, is less than or equal to 75 percent of the interface bandwidth. If the new aggregate bandwidth amount exceeds 75 percent of the interface bandwidth, the policy map is not modified.

Examples

The following example attaches the service policy map called policy9 to the input interface Serial1:

```
interface Serial1
service-policy input policy9
```

The following example attaches the policy called policy9 to the output interface serial1 to specify the service policy for the interface and enable CBWFQ on it:

```
interface serial1
service-policy output policy9
```

Related Commands

Command	Description
policy-map	Specifies a policy map to be assigned to an interface.
show policy	Displays configuration of all classes for a policy map.

show policy

To display the configuration of all classes comprising the specified service policy map or all classes for all existing policy maps, use the **show policy** global configuration command.

show policy *policy-map*

Syntax Description	<i>policy-map</i>	(Optional) The name of the service policy map whose complete configuration is to be displayed.
---------------------------	-------------------	--

Defaults All existing policy map configurations are displayed.

Command Modes Global configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines The **show policy** command displays the configuration of a service policy map created using the **policy-map** command. You can use the **show policy** command to display all class configurations comprising any existing service policy map, whether or not that service policy map has been attached to an interface.

Examples The following example displays the contents of the pol1 service policy map:

```
Router# show policy pol
Policy Map pol
  Weighted Fair Queueing
    Class class1
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class2
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class3
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class4
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class5
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class6
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class7
```

```

    Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class8
    Bandwidth 937 (kbps) Max thresh 64 (packets)

```

The following example displays the contents of all policy maps on the router:

```

Router# show policy
Policy Map poH1
  Weighted Fair Queueing
    Class class1
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class2
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class3
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class4
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class5
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class6
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class7
      Bandwidth 937 (kbps) Max thresh 64 (packets)
    Class class8
      Bandwidth 937 (kbps) Max thresh 64 (packets)
Policy Map policy2
  Weighted Fair Queueing
    Class class1
      Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class2
      Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class3
      Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class4
      Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class5
      Bandwidth 300 (kbps) Max thresh 64 (packets)
    Class class6
      Bandwidth 300 (kbps) Max thresh 64 (packets)

```

Related Commands	Command	Description
	show policy class	Displays the configuration for the specified class of the specified service policy map.
	show policy interface	Displays the configuration of all classes configured for all service policy maps on the specified interface.

show policy-map class

To display the configuration for the specified class of the specified policy map, use the **show policy class** global configuration command.

show policy *policy-map* **class** *class-name*

Syntax Description		
	<i>policy-map</i>	The name of a policy map that contains the class configuration to be displayed.
	<i>class-name</i>	The name of the class whose configuration is to be displayed.

Defaults No default behavior or values.

Command Modes Global configuration

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines You can use the **show policy class** command to display any one class configuration for any service policy map, whether or not the specified service policy map has been attached to an interface.

Examples The following example displays configurations for the class called class7 that belongs to the policy map pol.

```
Router# show policy pol class class7
Class class7
  Bandwidth 937 (kbps) Max Thresh 64 (packets)
```


Related Commands	Command	Description
	show policy	Displays the configuration for all classes of the specified service policy map.
	show policy interface	Displays the configuration of all classes configured for all service policies on the specified interface.

show policy-map interface

To display the configuration of all classes configured for all service policies on the specified interface, use the **show policy interface** global configuration command.

show policy-map interface *interface-name*

Syntax Description	<i>interface-name</i>	Name of the interface whose policy configuration is to be displayed.
---------------------------	-----------------------	--

Defaults	No default behavior.
-----------------	----------------------

Command Modes	Global configuration
----------------------	----------------------

Command History	Release	Modification
	12.0(5)T	This command was first introduced.
	12.0(5)XE	This command was introduced in Release 12.0 XE train.
	12.0(7)S	This command was introduced in Release 12.0 S train.
	12.1(1)E	This command was introduced in Release 12.1 E.

Usage Guidelines	The show policy interface command displays the configuration for classes on the specified interface only if a service policy has been attached to the interface.
-------------------------	---

Examples

The following example displays configurations for classes on the output interface e1/1:

```
Router# show policy interface e1/1
Ethernet1/1 output : pol
Weighted Fair Queueing
Class class1
  Output Queue: Conversation 264
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11548/0/0
Class class2
  Output Queue: Conversation 265
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11546/0/0
Class class3
  Output Queue: Conversation 266
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11546/0/0
Class class4
  Output Queue: Conversation 267
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11702/0/0
Class class5
  Output Queue: Conversation 268
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11701/0/0
Class class6
  Output Queue: Conversation 269
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11702/0/0
Class class7
  Output Queue: Conversation 270
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11857/0/0
Class class8
  Output Queue: Conversation 271
  Bandwidth 937 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 11858/1/0
```

Related Commands

Command	Description
show policy	Displays the configuration for all classes of the specified service policy map.
show policy class	Displays the configuration for the specified class of the specified service policy map.

■ show policy-map interface