



# MQC-Based Frame Relay Traffic Shaping

---

**Last Updated: October 6, 2011**

The MQC-Based Frame Relay Traffic Shaping feature provides users with the ability to configure Frame Relay traffic shaping (FRTS) using modular quality of service (QoS) command-line interface (CLI) commands. Modular QoS CLI is known as MQC.

This feature is available for the Cisco routers specified in the [MQC-Based Frame Relay Traffic Shaping, page 1](#) section.

---

## Supported Platforms

---

Cisco 1700 series, Cisco 2500 series, Cisco 2600 series, Cisco 3620 router, Cisco 3631 router, Cisco 3640 router, Cisco 3660 router, Cisco 3725 router, Cisco 3745 router, Cisco 7200 series, Cisco 7400 series, Cisco 7500 series and above

---

- [Finding Feature Information, page 1](#)
- [Prerequisites for MQC-Based Frame Relay Traffic Shaping, page 2](#)
- [Restrictions for MQC-Based Frame Relay Traffic Shaping, page 2](#)
- [Information About MQC-Based Frame Relay Traffic Shaping, page 2](#)
- [How to Configure MQC-Based Frame Relay Traffic Shaping, page 4](#)
- [Configuration Examples for MQC-Based Frame Relay Traffic Shaping, page 16](#)
- [Additional References, page 17](#)
- [Feature Information for MQC-Based Frame Relay Traffic Shaping, page 18](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.



---

**Americas Headquarters:**  
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

## Prerequisites for MQC-Based Frame Relay Traffic Shaping

Every permanent virtual circuit (PVC) to which FRTS using MQC is applied must have Frame Relay encapsulation enabled on the associated interface with the **encapsulation frame-relay** command.

**Note**

For FRTS using MQC for the routers specified in the [Prerequisites for MQC-Based Frame Relay Traffic Shaping, page 2](#) section, distributed Cisco Express Forwarding is not required. For FRTS using MQC for routers in the Cisco 7500 and above product range, distributed Cisco Express Forwarding is required.

## Restrictions for MQC-Based Frame Relay Traffic Shaping

You must create a default class for the service policy as specified with the **class class-default** command to configure FRTS using MQC on a Frame Relay PVC. The default class will have all the FRTS parameters applied to it. Refer to the [Creating a Policy Map and Entering CBWFQ Parameters for the Class Map, page 6](#) section for more information on creating a default class structure for a service policy.

If FRTS and fragmentation are applied to a PVC using MQC, the interface queue will change to dual first-in, first-out (FIFO) queueing. The two queues will consist of a high-priority queue to carry VoIP and certain control packets, and a low-priority queue to carry all other packets.

**Note**

In configurations created by using traditional FRTS commands, the minimum acceptable outgoing committed information rate (minCIR) will be used as the total available bandwidth for a policy map that has class-based weighted fair queueing (CBWFQ) attached to the map class for the PVC. If the MQC-Based Frame Relay Traffic Shaping feature is used to configure FRTS, the shaping rate that was configured in the parent policy map using MQC will be used as the total available bandwidth for the child policy map, if CBWFQ is configured. If both the **shape average** and **shape adaptive** commands are used for traffic shaping, the available bandwidth will be based on the parameters specified by the **shape adaptive** command.

## Information About MQC-Based Frame Relay Traffic Shaping

- [MQC Overview, page 2](#)
- [FRTS Overview, page 3](#)
- [Feature Design of MQC-Based Frame Relay Traffic Shaping, page 3](#)
- [Benefits of MQC-Based Frame Relay Traffic Shaping, page 4](#)

## MQC Overview

MQC is used to configure FRTS. MQC is a framework that provides a clear separation between a classification policy and the specification of other parameters that act on the results of that applied classification policy.

Before this feature was introduced in Cisco IOS Release 12.2(13)T, FRTS for the Cisco routers specified in the [MQC Overview, page 2](#) section could be configured only by using traditional FRTS commands (refer

to the [How to Configure MQC-Based Frame Relay Traffic Shaping, page 4](#) section for more information). With the addition of the MQC-Based Frame Relay Traffic Shaping feature, FRTS can be configured on routers throughout the Cisco router product line by using MQC.

Broadly, MQC is configured and implemented as follows:

- Define a traffic class with the **class-map** command.
- Create a service policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attach the service policy to the interface with the **service-policy** command.

For more detailed information on MQC, refer to the document Modular Quality of Service Command-Line Interface. MQC commands used for FRTS are further explained in the [How to Configure MQC-Based Frame Relay Traffic Shaping, page 4](#) section of this document.

## FRTS Overview

FRTS allows you to control the traffic going out through a PVC in order to match its flow to the speed of the remote target interface and to ensure that the traffic conforms to the parameters that have been set for it. Traffic that matches a particular profile can be shaped to meet downstream requirements, thereby eliminating the bottlenecks that occur in topologies that have data-rate mismatches.

The primary reasons you would use FRTS are the following:

- To allow high-priority packets to take precedence over other packets as they are encapsulated and forwarded over the Frame Relay network. FRTS is useful for applications such as VoIP and streaming video, which require a low latency to be effective.
- To control access to available bandwidth.
- To ensure that traffic conforms to the parameters established for it.
- To regulate the flow of traffic in order to avoid congestion that can occur when the sent traffic exceeds the access speed of its remote target interface.
- To eliminate bottlenecks in Frame Relay networks that have high-speed connections at the central site and low-speed connections at branch sites by configuring rate enforcement to limit the rate at which data is sent on the virtual circuit (VC) at the central site. Rate enforcement is a peak rate configured to limit outbound traffic.

## Feature Design of MQC-Based Frame Relay Traffic Shaping

The MQC-Based Frame Relay Traffic Shaping feature allows the Cisco routers specified in the [Feature Design of MQC-Based Frame Relay Traffic Shaping, page 3](#) section to have FRTS configured using MQC instead of traditional FRTS commands.

Before this feature was introduced in Cisco IOS Release 12.2(13)T, FRTS for the Cisco routers specified in the [Feature Design of MQC-Based Frame Relay Traffic Shaping, page 3](#) section could be configured only by using traditional FRTS commands (for example, the **frame-relay traffic-shaping** command). For traditional FRTS, all traffic shaping and fragmentation values are entered under the map class. Traffic shaping is defined by entering the **map-class frame-relay** command, then entering the traffic shaping and, optionally, fragmentation values.

For routers in the Cisco 7500 and above product range, Distributed Traffic Shaping (DTS) is used for traffic shaping. With DTS, the traffic-shaping values are configured by entering the **policy-map** command, then entering the traffic-shaping values. However, fragmentation values are still entered under the map class--the **map-class frame-relay** command is still used before any fragmentation values are entered.

The traffic-shaping commands supported by the MQC-Based Frame Relay Traffic Shaping feature are listed in the [Supported MQC Frame Relay Traffic Shaping Commands, page 4](#) section.

CBWFQ can also be configured under the policy map by entering the **policy-map** command and then entering the traffic-shaping CBWFQ values.

**Note**

Configuring traffic shaping using MQC and configuring traffic shaping using traditional FRTS commands are mutually exclusive. Traffic shaping cannot be configured on the same interface using both methods.

- [Supported MQC Frame Relay Traffic Shaping Commands, page 4](#)

## Supported MQC Frame Relay Traffic Shaping Commands

The following MQC traffic-shaping commands are supported by the MQC-Based Frame Relay Traffic Shaping feature:

- **shape {average | peak}**
- **shape adaptive**
- **shape fecn-adapt**
- **shape max-buffers**

The **frame-relay ip rtp priority** command is not supported.

**Note**

Before this feature was introduced in Cisco IOS Release 12.2(13)T, FRTS for the Cisco routers specified in the [Supported MQC Frame Relay Traffic Shaping Commands, page 4](#) section could be configured only by using traditional FRTS commands (refer to the [How to Configure MQC-Based Frame Relay Traffic Shaping, page 4](#) section for more information).

## Benefits of MQC-Based Frame Relay Traffic Shaping

MQC allows users to specify a traffic class independently of QoS parameters.

The MQC-Based Frame Relay Traffic Shaping feature allows users to apply FRTS parameters using MQC across the entire Cisco router product line.

This feature ensures that FRTS is defined in the same manner for routers across the Cisco router product line, rather than only for routers in the Cisco 7500 and above product range.

Before this feature was introduced in Cisco IOS Release 12.2(13)T, FRTS for the Cisco routers specified in the [Benefits of MQC-Based Frame Relay Traffic Shaping, page 4](#) section could be defined only by using traditional FRTS commands. Using different methods to define FRTS for different routers can introduce inconsistency and complexity when FRTS is being implemented on different router platforms.

## How to Configure MQC-Based Frame Relay Traffic Shaping

- [Creating a Class Map and Specifying Match Criteria for CBWFQ, page 5](#)
- [Creating a Policy Map and Entering CBWFQ Parameters for the Class Map, page 6](#)
- [Creating a Shaping Policy Map and Entering FRTS Values for the Default Class Map, page 7](#)

- [Attaching the Class-Based Weighted Fair Queueing Policy Map to the Shaping Policy Map, page 9](#)
- [Specifying a Map Class and Attaching a Service Policy for the PVC, page 10](#)
- [Configuring an Interface or Subinterface for Frame Relay and Associating a Map Class with a PVC, page 11](#)
- [Defining Fragmentation Parameters for the PVC, page 13](#)
- [Adding the Policy Map to the Map Class, page 14](#)

## Creating a Class Map and Specifying Match Criteria for CBWFQ

To create a class map and specify match criteria for CBWFQ, use the following commands.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match** *match-criteria*
5. **end**

### DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <b>enable</b></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<p><b>Step 2</b> <b>configure terminal</b></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <b>class-map</b> <i>class-map-name</i></p> <p><b>Example:</b></p> <pre>Router(config)# class-map voice</pre>	<p>Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode.</p> <ul style="list-style-type: none"> <li>• The example command creates a class map named "voice".</li> </ul>
<p><b>Step 4</b> <b>match</b> <i>match-criteria</i></p> <p><b>Example:</b></p> <pre>Router(config-cmap)# match ip dscp ef</pre>	<p>Identifies packets that will belong to the class map.</p> <ul style="list-style-type: none"> <li>• The example command identifies an IP differentiated service code point (DSCP) value of EF (101110) as a match criterion for the class map named "voice".</li> </ul>

Command or Action	Purpose
<b>Step 5</b> <code>end</code>  <b>Example:</b>  <code>Router(config-cmap)# end</code>	(Optional) Exits class-map configuration mode and returns to privileged EXEC mode.

## Creating a Policy Map and Entering CBWFQ Parameters for the Class Map

To create a policy map and enter CBWFQ parameters for the class map, use the following commands.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `policy-map policy-map-name`
4. `class name`
5. `priority [bandwidth-kbps | percent percentage] [burst]`
6. `end`

### DETAILED STEPS

Command or Action	Purpose
<b>Step 1</b> <code>enable</code>  <b>Example:</b>  <code>Router&gt; enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b> <code>configure terminal</code>  <b>Example:</b>  <code>Router# configure terminal</code>	Enters global configuration mode.
<b>Step 3</b> <code>policy-map <i>policy-map-name</i></code>  <b>Example:</b>  <code>Router(config)# policy-map llq</code>	Creates a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode. <ul style="list-style-type: none"> <li>• The example command creates a policy map named "llq". This policy map will be used for low latency queueing (LLQ).</li> </ul>

Command or Action	Purpose
<p><b>Step 4</b> <code>class name</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap)# class voice</pre>	<p>Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <li>The example command creates a traffic class named "voice".</li> </ul>
<p><b>Step 5</b> <code>priority [bandwidth-kbps   percent percentage] [burst</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# priority 32</pre>	<p>(Optional) Gives priority to a class of traffic belonging to a policy map.</p> <ul style="list-style-type: none"> <li>The example command provides a guaranteed allowed bandwidth of 32 kbps and a guaranteed low latency for up to 32 kbps to the traffic class "voice" that was created in <a href="#">Creating a Policy Map and Entering CBWFQ Parameters for the Class Map</a>, page 6.</li> </ul>
<p><b>Step 6</b> <code>end</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# end</pre>	<p>(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.</p>

## Creating a Shaping Policy Map and Entering FRTS Values for the Default Class Map

To create a shaping policy map and enter FRTS values for the default class map, use the following commands.

### SUMMARY STEPS

- `enable`
- `configure terminal`
- `policy-map policy-map-name`
- `class class-default`
- `shape {average | peak} {mean-rate [burst-size [excess-burst-size]] | percent percentage [burst-size ms [excess-burst [ms]]]}`
- `shape adaptive mean-rate-lower-bound`
- `shape fecn-adapt`
- `end`

## DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <code>enable</code></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <code>policy-map <i>policy-map-name</i></code></p> <p><b>Example:</b></p> <pre>Router(config)# policy-map shape-policy-map</pre>	<p>Creates a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode.</p> <ul style="list-style-type: none"> <li>The example command creates a policy map named shape-policy-map.</li> </ul>
<p><b>Step 4</b> <code>class class-default</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap)# class class-default</pre>	<p>Specifies the default class (commonly known as the class-default class) before you configure its policy and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <li>The class-default class is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes whose policy is defined in the policy map.</li> <li>Only one class is defined with this command (the class-default class); therefore, it will match all traffic.</li> </ul>
<p><b>Step 5</b> <code>shape {average   peak} {<i>mean-rate</i> [<i>burst-size</i> [<i>excess-burst-size</i>]]   percent <i>percentage</i> [<i>burst-size ms</i> [<i>excess-burst ms</i>]]}</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# shape average 64000</pre>	<p>Shapes traffic to the indicated bit rate according to the algorithm specified.</p> <ul style="list-style-type: none"> <li>The example command configures a shape entity with a CIR of 64,000 bytes per second (bps).</li> <li>Any FRTS commands supported by this feature can be used for traffic shaping. Refer to the <a href="#">Restrictions for MQC-Based Frame Relay Traffic Shaping, page 2</a> section for a list of supported FRTS commands.</li> </ul>
<p><b>Step 6</b> <code>shape adaptive <i>mean-rate-lower-bound</i></code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# shape adaptive 32000</pre>	<p>(Optional) Configures a Frame Relay PVC to estimate the available bandwidth by backward explicit congestion notification (BECN) integration while traffic shaping is enabled.</p> <ul style="list-style-type: none"> <li>The example command sets the lower bound CIR to 32,000 bps when BECNs are received.</li> </ul>



Command or Action	Purpose
<b>Step 7</b> <b>shape fecn-adapt</b>  <b>Example:</b>  <pre>Router(config-pmap-c)# shape fecn-adapt</pre>	(Optional) Configures a Frame Relay interface to reflect received forward explicit congestion notification (FECN) bits as BECN bits in Q.922 "Test Response" messages. <ul style="list-style-type: none"> <li>The example command adapts the Frame Relay message with FECN to BECN.</li> </ul>
<b>Step 8</b> <b>end</b>  <b>Example:</b>  <pre>Router(config-pmap-c)# end</pre>	(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.

## Attaching the Class-Based Weighted Fair Queueing Policy Map to the Shaping Policy Map

To attach the CBWFQ policy map to the shaping policy map, use the following commands.

### SUMMARY STEPS

- enable
- configure terminal
- policy-map *policy-map-name*
- class class-default
- service-policy *policy-map-name*
- end

### DETAILED STEPS

Command or Action	Purpose
<b>Step 1</b> <b>enable</b>  <b>Example:</b>  <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b> <b>configure terminal</b>  <b>Example:</b>  <pre>Router# configure terminal</pre>	Enters global configuration mode.

Command or Action	Purpose
<p><b>Step 3</b> <code>policy-map <i>policy-map-name</i></code></p> <p><b>Example:</b></p> <pre>Router(config)# policy-map shape-policy-map</pre>	<p>Creates a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode.</p> <ul style="list-style-type: none"> <li>The example command creates a policy map named shape-policy-map.</li> </ul>
<p><b>Step 4</b> <code>class class-default</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap)# class class-default</pre>	<p>Specifies the default class (commonly known as the class-default class) before you configure its policy and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <li>The class-default class is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes whose policy is defined in the policy map.</li> <li>Only one class is defined with this command (the class-default class); therefore, it will match all traffic.</li> </ul>
<p><b>Step 5</b> <code>service-policy <i>policy-map-name</i></code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# service-policy llq</pre>	<p>Attaches a service policy as a traffic shaping policy within a policy map.</p> <ul style="list-style-type: none"> <li>This parent-child service policy association is called a hierarchical service policy.</li> <li>The example command creates a hierarchical service policy with the parent policy map being "shape-policy-map", and the child policy map being "llq".</li> </ul>
<p><b>Step 6</b> <code>end</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# end</pre>	<p>(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.</p>

## Specifying a Map Class and Attaching a Service Policy for the PVC

To specify a map class and attach a service policy for the PVC, use the following commands.

### SUMMARY STEPS

- `enable`
- `configure terminal`
- `map-class frame-relay map-class-name`
- `service-policy {input | output} policy-map-name`
- `end`

## DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <code>enable</code></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <code>map-class frame-relay map-class-name</code></p> <p><b>Example:</b></p> <pre>Router (config)# map-class frame-relay shape-map-class</pre>	<p>Specifies a map class to define fragmentation values for both a PVC and a MQC policy map attachment and enters static maps class configuration mode.</p> <ul style="list-style-type: none"> <li>The example command specifies a map class named <code>shape-map-class</code>.</li> </ul>
<p><b>Step 4</b> <code>service-policy {input   output} policy-map-name</code></p> <p><b>Example:</b></p> <pre>Router(config-map-class)# service-policy input shape-policy-map</pre>	<p>Attaches a service policy for a PVC.</p>
<p><b>Step 5</b> <code>end</code></p> <p><b>Example:</b></p> <pre>Router(config-map-class)# end</pre>	<p>(Optional) Exits static maps class configuration mode and returns to privileged EXEC mode.</p>

## Configuring an Interface or Subinterface for Frame Relay and Associating a Map Class with a PVC

To configure an interface or subinterface for Frame Relay and associate a map class with a PVC, use the following commands.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation frame-relay** [**MFR** | **ietf**]
5. **exit**
6. **interface** *type slot / port . subinterface-number* [**multipoint** | **point-to-point**]
7. **ip address** *ip-address mask* [**secondary**]
8. **frame-relay interface-dlci** *dlci* [**ietf** | **cisco**]
9. **class** *name*
10. **end**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>type number</i> [ <i>name-tag</i> ]  <b>Example:</b> Router(config)# interface serial 0/0	Configures an interface type and enters interface configuration mode.
<b>Step 4</b>	<b>encapsulation frame-relay</b> [ <b>MFR</b>   <b>ietf</b> ]  <b>Example:</b> Router(config-if)# encapsulation frame-relay	Enables Frame Relay encapsulation.
<b>Step 5</b>	<b>exit</b>  <b>Example:</b> Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	<b>interface</b> <i>type slot / port . subinterface-number</i> [multipoint   point-to-point]  <b>Example:</b>  <pre>Router(config)# interface serial 0/0.1 point-to-point</pre>	Configures a subinterface and enters subinterface configuration mode.
Step 7	<b>ip address</b> <i>ip-address mask</i> [secondary]  <b>Example:</b>  <pre>Router(config-subif)# ip address 192.168.1.1 255.255.255.0</pre>	Sets the primary IP address and mask for the subinterface.
Step 8	<b>frame-relay interface-dlci</b> <i>dlci</i> [ietf   cisco]  <b>Example:</b>  <pre>Router(config-subif)# frame-relay interface-dlci 100</pre>	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on the router and enters Frame Relay DLCI configuration mode. <ul style="list-style-type: none"> <li>The example command creates a PVC with a DLCI number of 100 under serial 0/0.1 subinterface.</li> </ul>
Step 9	<b>class</b> <i>name</i>  <b>Example:</b>  <pre>Router(config-fr-dlci)# class shape-map-class</pre>	Associates a map class with the subinterface. <ul style="list-style-type: none"> <li>The example command associates the map class named <code>shape-map-class</code> with the PVC created in <a href="#">Configuring an Interface or Subinterface for Frame Relay and Associating a Map Class with a PVC</a>, page 11.</li> </ul>
Step 10	<b>end</b>  <b>Example:</b>  <pre>Router(config-fr-dlci)# end</pre>	(Optional) Exits Frame Relay DLCI configuration mode and returns to privileged EXEC mode.

## Defining Fragmentation Parameters for the PVC

To define fragmentation parameters for the PVC, use the following commands.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **map-class frame-relay** *map-class-name*
4. **frame-relay fragment** *fragment-size* [switched]
5. **end**

## DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <code>enable</code></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <code>map-class frame-relay map-class-name</code></p> <p><b>Example:</b></p> <pre>Router (config)# map-class frame-relay shape-map-class</pre>	<p>Specifies a map class to define fragmentation parameters for both a PVC and an MQC policy map attachment and enters static map class configuration mode.</p> <ul style="list-style-type: none"> <li>The example command specifies a map class named <code>shape-map-class</code>.</li> </ul>
<p><b>Step 4</b> <code>frame-relay fragment fragment-size [switched]</code></p> <p><b>Example:</b></p> <pre>Router(config-map-class)# frame-relay fragment 80</pre>	<p>(Optional) Enables fragmentation of Frame Relay frames for a Frame Relay map class.</p> <ul style="list-style-type: none"> <li>The example command specifies that 80 payload bytes from the original Frame Relay frame will go into each fragment.</li> </ul>
<p><b>Step 5</b> <code>end</code></p> <p><b>Example:</b></p> <pre>Router(config-map-class)# end</pre>	<p>(Optional) Exits static maps class configuration mode and returns to privileged EXEC mode.</p>

## Adding the Policy Map to the Map Class

To add the policy map to the map class, use the following commands.

### SUMMARY STEPS

- `enable`
- `configure terminal`
- `policy-map policy-map-name`
- `class class-default`
- `service-policy policy-map`
- `end`

## DETAILED STEPS

Command or Action	Purpose
<p><b>Step 1</b> <code>enable</code></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<p><b>Step 2</b> <code>configure terminal</code></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 3</b> <code>policy-map <i>policy-map-name</i></code></p> <p><b>Example:</b></p> <pre>Router(config)# policy-map shape-policy-map</pre>	<p>Creates a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode.</p> <ul style="list-style-type: none"> <li>The example command creates a policy map named shape-policy-map.</li> </ul>
<p><b>Step 4</b> <code>class class-default</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap)# class class-default</pre>	<p>Specifies the default class (commonly known as the class-default class) before you configure its policy and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <li>The class-default class is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes whose policy is defined in the policy map.</li> <li>Only one class is defined with this command (the class-default class); therefore, it will match all traffic.</li> </ul>
<p><b>Step 5</b> <code>service-policy <i>policy-map</i></code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# service-policy shape-policy-map</pre>	<p>Adds a policy map to a map class and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <li>The example command attaches the policy map named shape-policy-map. This command will also attach the policy map shape to any other PVCs that are using the map class.</li> </ul>
<p><b>Step 6</b> <code>end</code></p> <p><b>Example:</b></p> <pre>Router(config-pmap-c)# end</pre>	<p>(Optional) Exits policy-map class configuration mode and returns to privileged EXEC mode.</p>

## Configuration Examples for MQC-Based Frame Relay Traffic Shaping

- [Example Configuring Class-Based Weighted Fair Queueing, page 16](#)
- [Example Configuring Class-Based Weighted Fair Queueing with Fragmentation, page 16](#)

### Example Configuring Class-Based Weighted Fair Queueing

The following example provides a sample configuration for Class-Based Weighted Fair Queueing (CBWFQ) with FRTS:

```
class-map voice
 match ip dscp ef
policy-map llq
 class voice
  priority 32
policy-map shape-policy-map
 class class-default
  shape average 64000
  shape adaptive 32000
  service-policy llq
map-class frame-relay shape-map-class

service-policy output shape-policy-map
interface serial 0/0
 encapsulation frame-relay
interface serial 0/0.1 point-to-point
 ip address 192.168.1.1 255.255.255.0
 frame-relay interface-dlci 100
 class shape-map-class
```

### Example Configuring Class-Based Weighted Fair Queueing with Fragmentation

The following example provides a sample configuration for CBWFQ and fragmentation with FRTS. This configuration example is exactly the same as the example shown in the Example Configuring Class-Based Weighted Fair Queueing section, with the addition of the **frame-relay fragment** command to configure fragmentation.

```
class-map voice
 match ip dscp ef
policy-map llq
 class voice
  priority 32
policy-map shape-policy-map
 class class-default
  shape average 64000
  shape adaptive 32000
  service-policy llq
map-class frame-relay shape-map-class
 frame-relay fragment 80
service-policy output shape-policy-map
interface serial 0/0
 encapsulation frame-relay
interface serial 0/0.1 point-to-point
 ip address 192.168.1.1 255.255.255.0
 frame-relay interface-dlci 100
 class shape-map-class
```



## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
WAN commands	<i>Cisco IOS Wide Area Network Command Reference</i>
MQC commands	Modular Quality of Service Command-Line Interface

### Standards

Standard	Title
None	--

### MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

### Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

# Feature Information for MQC-Based Frame Relay Traffic Shaping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 1** Feature Information for MQC-Based Frame Relay Traffic Shaping

Feature Name	Releases	Feature Information
MQC-Based Frame Relay Traffic Shaping	12.2(13)T	<p>The MQC-Based Frame Relay Traffic Shaping feature provides users with the ability to configure Frame Relay traffic shaping (FRTS) using modular quality of service (QoS) command-line interface (CLI) commands. Modular QoS CLI is known as MQC.</p> <p>The following commands were introduced or modified: <b>shape adaptive</b>, <b>shape fecn-adapt</b>.</p>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2011 Cisco Systems, Inc. All rights reserved.