Chapter 16

Rewriting Packet Header Information

As packets enter or exit a network, edge routers might be required to alter the class-of-service (CoS) settings of the packets. Rewrite rules set the value of the CoS bits within the packet's header. Each rewrite rule reads the current forwarding class and loss priority information associated with the packet, locates the chosen CoS value from a table, and writes this CoS value into the packet header.

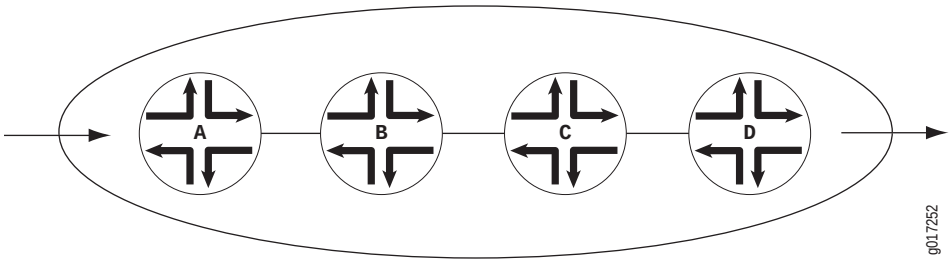
In effect, the rewrite rule performs the opposite function of the behavior aggregate (BA) classifier used when the packet enters the router. As the packet leaves the routing platform, the final CoS action is generally the application of a rewrite rule.

You configure rewrite rules to alter CoS values in outgoing packets on the outbound interfaces of an edge router to meet the policies of a targeted peer. This allows the downstream router in a neighboring network to classify each packet into the appropriate service group.

In addition, you often need to rewrite a given marker (IP precedence, DSCP, IEEE 802.1P, or MPLS EXP settings) at the inbound interfaces of an edge router to accommodate BA classification by core devices.

In Figure 12, Router A rewrites the CoS bits in incoming packet to accommodate BA classification performed by Routers B and C. Router D alters the CoS bits of the packets before transmitting them to the neighboring network.

Figure 12: Packet Flow Across the Network



To configure CoS rewrite rules, you define the rewrite rule and apply it to an interface. You include the following statements at the [edit class-of-service] hierarchy level:

```
class-of-service {
    interfaces {
        interface-name {
            unit logical-unit-number {
                rewrite-rules {
                    dscp (rewrite-name | default);
                    dscp-ipv6 (rewrite-name | default);
                    exp (rewrite-name | default) protocol protocol-types;
                    exp-push-push-push default;
                    exp-swap-push-push default;
                    frame-relay-de (rewrite-name | default);
                    ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
                    inet-precedence (rewrite-name | default);
                }
            }
        }
    }
    rewrite-rules {
        (dscp | dscp-ipv6 | exp | frame-relay-de | ieee-802.1 | inet-precedence)
            rewrite-name {
            import (rewrite-name | default);
            forwarding-class class-name {
                loss-priority level code-point (alias | bits);
            }
        }
    }
}
```

This chapter discusses the following topics:

Applying a Default Rewrite Rule

By default, rewrite rules are not applied to interfaces. If you want to apply a rewrite rule, you can either design your own rule and apply it to an interface, or you can apply a default rewrite rule. To apply default rewrite rules, include one or more of the following statements at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
dscp default;
dscp-ipv6 default;
exp default;
ieee-802.1 default vlan-tag (outer | outer-and-inner);
inet-precedence default;
```

Table 32 shows the default rewrite rule mappings. These are based on the default bit definitions of DSCP, DSCP IPv6, EXP, IEEE, and IP CoS values, as shown in Table 8 on page 38, and the default forwarding classes shown in Table 16 on page 81.

When the software detects packets whose CoS values match the forwarding class and PLP values in the first two columns in Table 32, the software maps the header bits of those packets to the code-point aliases in the last column in Table 32. The code-point aliases in the last column map to the CoS bits shown in Table 8 on page 38.

Table 32: Default Packet Header Rewrite Mappings

Map from Forwarding Class	PLP Value	Map to DSCP/DSCP IPv6/ EXP/IEEE/IP
expedited-forwarding	low	ef
expedited-forwarding	high	ef
assured-forwarding	low	af11
assured-forwarding	high	af12 (DSCP/DSCP IPv6/EXP)
best-effort	low	be
best-effort	high	be
network-control	low	nc1/cs6
network-control	high	nc2/cs7

In the following example, the so-1/2/3.0 interface is assigned the default DSCP rewrite rule. One result of this configuration is that each packet exiting the interface with the expedited-forwarding forwarding class and the high or low loss priority has its DSCP bits rewritten to the DSCP ef code-point alias. Table 8 on page 38 shows that this code-point alias maps to the 101110 bits.

Another result of this configuration is that all packets exiting the interface with the best-effort forwarding class and the high or low loss priority have their EXP bits rewritten to the EXP be code-point alias. Table 8 on page 38 shows that this code-point alias maps to the 000 bits.

To evaluate all the implications of this example, see Table 8 on page 38 and Table 32 on page 209.

```
class-of-service {
    interfaces {
        so-1/2/3 {
            unit 0 {
                rewrite-rules {
                    dscp default;
                }
            }
        }
    }
}
```

Configuring Rewrite Rules

You define markers in the rewrite rules section of the CoS configuration hierarchy and reference them in the logical interface configuration. This model supports marking on the DSCP, DSCP IPv6, IP precedence, IEEE 802.1, and MPLS EXP CoS values.

To configure a rewrite-rules mapping and associate it with the appropriate forwarding class and code-point alias or bit set, include the rewrite-rules statement at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
rewrite-rules {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (alias | bits);
        }
    }
}
```

The rewrite rule sets the code-point aliases and bit patterns for a specific forwarding class and PLP. The inputs for the map are the forwarding class and the PLP. The output of the map is the code-point alias or bit pattern. For more information about how CoS maps work, see Table 5 on page 9.

By default, IP precedence rewrite rules alter the first three bits on the type of service (TOS) byte while leaving the last three bits unchanged. This default behavior is not configurable. The default behavior applies to rules you configure by including the inet-precedence statement at the [edit class-of-service rewrite-rules] hierarchy level. The default behavior also applies to rewrite rules you configure for MPLS packets with IPv4 payloads. You configure these types of rewrite rules by including the mpls-inet-both or mpls-inet-both-non-vpn option at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol] hierarchy level.

Bits Preserved, Cleared, and Rewritten

For every incoming packet, the ingress classifier decodes the ingress CoS bits into a forwarding class and packet loss priority (PLP) combination. The egress CoS information depends on which type of rewrite marker is active, as follows:

- For Multiprotocol Label Switching (MPLS) EXP and IEEE 802.1 rewrite markers, values are derived from the forwarding class and PLP values in rewrite rules. MPLS EXP and IEEE 802.1 markers are not preserved because they are part of the Layer 2 encapsulation.

- For IP precedence and DiffServ code point (DSCP) rewrite markers, the marker alters the first three bits on the type-of-service (ToS) byte while leaving the last three bits unchanged.

Assigning the Rewrite-Rules Configuration to the Output Logical Interface

To assign the rewrite-rules configuration to the output logical interface, include the rewrite-rules statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
rewrite-rules {
    dscp (rewrite-name | default);
    dscp-ipv6 (rewrite-name | default);
    exp (rewrite-name | default) protocol protocol-types;
    exp-push-push-push default;
    exp-swap-push-push default;
    ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
    inet-precedence (rewrite-name | default);
}
```

You can use interface wildcards for *interface-name* and *logical-unit-number*. You can also include Layer 2 and Layer 3 rewrite information in the same configuration.

NOTE: On M-series platforms only, if you include the control-word statement at the [edit protocols l2circuit neighbor *address* interface *interface-name*] hierarchy level, the software cannot rewrite MPLS EXP bits.

DSCP and DSCP IPv6 rewrite rules are not supported on IQ PICs installed on M320 and T-series platforms.

On T-series and M320 platforms, for a single interface, you cannot enable a rewrite rule on a subset of forwarding classes. You must assign a rewrite rule to either none of the forwarding classes or all of the forwarding classes. When you assign a rewrite rule to a subset of forwarding classes, the commit does not fail, and the subset of forwarding classes work as expected. However, the forwarding classes to which the rewrite rule is not assigned are rewritten to all zeros.

For example, if you configure a Differentiated Services code point (DSCP) rewrite rule, the bits in the forwarding classes to which you do not assign the rewrite rule are rewritten to 000000; if you configure an IP precedence rewrite rule, the bits in the forwarding classes to which you do not assign the rewrite rule are rewritten to 000.

Assigning the IEEE 802.1p Rewrite Rule to Dual VLAN Tags

By default, when you apply an IEEE 802.1p rewrite rule to an output logical interface, the software rewrites the IEEE bits in the outer VLAN tag only.

For Gigabit Ethernet IQ2 PICs and 10-Gigabit Ethernet IQ2 PICs only, you can rewrite the IEEE bits in both the outer and inner VLAN tags of the tagged Ethernet frames. When you enable the CoS rewrite for both tags, the same IEEE 802.1p rewrite table is used for the inner and outer VLAN tag.

To rewrite both the outer and inner VLAN tags, include the vlan-tag outer-and-inner statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1 (*rewrite-name* | default)] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules ieee-802.1 (rewrite-name | default)]
vlan-tag outer-and-inner;
```

To explicitly specify the default behavior, include the vlan-tag outer statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules ieee-802.1 (*rewrite-name* | default)] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules ieee-802.1 (rewrite-name | default)]
vlan-tag outer;
```

For more information about VLAN tags, see the *JUNOS Network Interfaces Configuration Guide*.

Example: Assigning the IEEE 802.1p Rewrite Rule to Dual VLAN Tags

Apply the ieee8021p-rwrule1 rewrite rule to both inner and outer VLAN tags of Ethernet-tagged frames exiting the ge-0/0/0.0 interface:

```
class-of-service {
    interfaces {
        ge-0/0/0 {
            unit 0 {
                rewrite-rules {
                    ieee-802.1 ieee8021p-rwrule1 vlan-tag outer-and-inner;
                }
            }
        }
    }
}
```

Rewriting EXP Bits on a Particular Node

To configure a custom table to rewrite the EXP bits, also known as CoS bits, on a particular node, the classifier table and the rewrite table must specify exactly the same CoS values.

In addition, the least significant bit of the CoS value itself must represent the PLP value. For example, CoS value 000 must be associated with PLP low, 001 must be associated with PLP high, and so forth.

Example: Rewriting EXP Bits on a Particular Node

Configure a custom table to rewrite the EXP bits on a particular node:

```
[edit class-of-service]
classifiers {
    exp exp-class {
        forwarding-class be {
            loss-priority low code-points 000;
            loss-priority high code-points 001;
        }
        forwarding-class af {
            loss-priority low code-points 010;
            loss-priority high code-points 011;
        }
        forwarding-class ef {
            loss-priority low code-points 100;
            loss-priority high code-points 101;
        }
        forwarding-class nc {
            loss-priority low code-points 110;
            loss-priority high code-points 111;
        }
    }
}
```

```
rewrite-rules {
    exp exp-rw {
        forwarding-class be {
            loss-priority low code-point 000;
            loss-priority high code-point 001;
        }
        forwarding-class af {
            loss-priority low code-point 010;
            loss-priority high code-point 011;
        }
        forwarding-class ef {
            loss-priority low code-point 100;
            loss-priority high code-point 101;
        }
        forwarding-class nc {
            loss-priority low code-point 110;
            loss-priority high code-point 111;
        }
    }
}
```

Rewriting MPLS and IPv4 Packet Headers

You can apply a rewrite rule to MPLS and IPv4 packet headers simultaneously. This allows you to initialize MPLS EXP and IP precedence bits at LSP ingress. You can configure different rewrite rules depending on whether the traffic is VPN or non-VPN.

The default MPLS EXP rewrite table contents are shown in Table 33.

Table 33: Default MPLS EXP Rewrite Table

Forwarding Class	Loss Priority	CoS Value
best-effort	low	000
best-effort	high	001
expedited-forwarding	low	010
expedited-forwarding	high	011
assured-forwarding	low	100
assured-forwarding	high	101
network-control	low	110
network-control	high	111

By default, IP precedence rewrite rules alter the first three bits on the type of service (TOS) byte while leaving the last three bits unchanged. This default behavior applies to rewrite rules you configure for MPLS packets with IPv4 payloads.

To override the default MPLS EXP rewrite table and rewrite MPLS and IPv4 packet headers simultaneously, include the protocol statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name*] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules exp rewrite-rule-name]
protocol protocol-types;
```

The protocol statement defines the types of MPLS packets and packet headers to which the specified rewrite rule is applied. The MPLS packet can be a standard MPLS packet or an MPLS packet with an IPv4 payload. Specify the type of MPLS packet using the following options:

- mpls-any—Applies the rewrite rule to MPLS packets and writes the CoS value to MPLS headers.
- mpls-inet-both—Applies the rewrite rule to VPN MPLS packets with IPv4 payloads. On M320 and T-series platforms, writes the CoS value to the MPLS and IPv4 headers. On M-series routing platforms, causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.
- mpls-inet-both-non-vpn—Applies the rewrite rule to non-VPN MPLS packets with IPv4 payloads. On M320 and T-series platforms, writes the CoS value to the MPLS and IPv4 headers. On M-series routing platforms, causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for the MPLS EXP value, and the configured rewrite code point for IP precedence.

An alternative to overwriting the default with a rewrite-rules mapping is to configure the default packet header rewrite mappings, as shown in Table 32 on page 209.

By default, IP precedence rewrite rules alter the first three bits on the TOS byte while leaving the last three bits unchanged. This default behavior is not configurable. The default behavior applies to rules you configure by including the inet-precedence statement at the [edit class-of-service rewrite-rules] hierarchy level. The default behavior also applies to rewrite rules you configure for MPLS packets with IPv4 payloads. You configure these types of rewrite rules by including the mpls-inet-both or mpls-inet-both-non-vpn option at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol] hierarchy level.

Example: Rewriting MPLS and IPv4 Packet Headers

On a M320 and T-series platform, configure rewrite tables and apply them in various ways to achieve the following results:

- For interface so-3/1/0, the three EXP rewrite tables are applied to packets, depending on the protocol of the payload:
  - IPv4 packets (VPN) that enter the LSPs on interface so-3/1/0 are initialized with values from rewrite table exp-inet-table. An identical three-bit value is written into the IP precedence and MPLS EXP bit fields.
  - IPv4 packets (non-VPN) that enter the LSPs on interface so-3/1/0 are initialized with values from rewrite table rule-non-vpn. An identical three-bit value is written into the IP precedence and MPLS EXP bit fields.
  - Non-IPv4 packets that enter the LSPs on interface so-3/1/0 are initialized with values from rewrite table rule1, and written into the MPLS EXP header field only. The statement exp rule1 has the same result as exp rule1 protocol mpls-any.
- For interface so-3/1/0, IPv4 packets transmitted over a non-LSP layer are initialized with values from IP precedence rewrite table rule2.
- For interface so-3/1/1, IPv4 packets that enter the LSPs are initialized with values from EXP rewrite table exp-inet-table. An identical 3-bit value is written into the IP precedence and MPLS EXP bit fields.
- For interface so-3/1/1, MPLS packets other than IPv4 Layer 3 types are also initialized with values from table exp-inet-table. For VPN MPLS packets with IPv4 payloads, the CoS value is written to MPLS and IPv4 headers. For VPN MPLS packets without IPv4 payloads, the CoS value is written to MPLS headers only.

```
[edit class-of-service]
rewrite-rules {
    exp exp-inet-table {
        forwarding-class best-effort {
            loss-priority low code-point 000;
            loss-priority high code-point 001;
        }
        forwarding-class assured-forwarding {
            loss-priority low code-point 010;
            loss-priority high code-point 011;
        }
        forwarding-class expedited-forwarding {
            loss-priority low code-point 111;
            loss-priority high code-point 110;
        }
        forwarding-class network-control {
            loss-priority low code-point 100;
            loss-priority high code-point 101;
        }
    }
    exp rule1 {
        ...
    }
```

```
        inet-precedence rule2 {
            ...
        }
    }
        exp rule_non_vpn {
            ...
    }

    interfaces {
        so-3/1/0 {
            unit 0 {
                rewrite-rules {
                    exp rule1;
                    inet-precedence rule2;
                    exp exp-inet-table protocol mpls-inet-both;# for all VPN traffic
                    exp rule_non_vpn protocol mpls-inet-both-non-vpn;
                                                            # for all non-VPN traffic
                }
            }
        }
        so-3/1/1 {
            unit 0 {
                rewrite-rules {
                    exp exp-inet-table protocol [mpls-any mpls-inet-both];
                }
            }
        }
    }
```

Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

In interprovider, carrier-of-carrier, and complex traffic engineering scenarios, it is sometimes necessary to push three labels on the next hop, using a swap-push-push or triple-push operation.

By default, on M-series routing platforms, the top MPLS EXP label of an outgoing packet is not rewritten when you configure swap-push-push and triple-push operations. On M-series routing platforms, you can rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining the CoS of an incoming MPLS or non-MPLS packet.

When the software performs a swap-push-push operation and no rewriting is configured, the EXP fields of all three labels are the same as in the old label. If there is EXP rewriting configured, the EXP bits of the bottom two labels are overwritten with the table entry. The EXP setting of the top label is retained even with rewriting.

To push three labels on all incoming MPLS packets, include the exp-swap-push-push default statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
exp-swap-push-push default;
```

When the software performs a push-push-push operation and if no rewriting is configured, the EXP fields of the bottom two labels are zero. If EXP rewriting is configured, the EXP fields of the bottom two labels are rewritten with the table entry's rewrite value. The EXP field of the top label is inserted with the Qn + PLP value. This Qn reflects the final classification by a multifield (MF) classifier if one exists, regardless of whether rewriting is configured.

To push three labels on incoming non-MPLS packets, include the **exp-push-push-push default** statement at the **[edit class-of-service interfaces** *interface-name* **unit** *logical-unit-number* **rewrite-rules]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
exp-push-push-push default;
```

These configurations apply the default MPLS EXP rewrite table, as shown in Table 33 on page 214. You can configure these operations and override the default MPLS EXP rewrite table with a custom table. For more information about writing and applying a custom rewrite table, see "Configuring Rewrite Rules" on page 210 and "Assigning the Rewrite-Rules Configuration to the Output Logical Interface" on page 211.

NOTE: With a three-label stack, if you do not include the **exp-swap-push-push default** or **exp-push-push-push default** statement in the configuration, the top label's EXP bits are set to zero.

Example: Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

Configure a swap-push-push operation, and override the default rewrite table with a custom table:

```
[edit class-of-service]
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
interfaces {
    so-1/1/3 {
      unit 0 {
        rewrite-rules {
          exp exp_rew;   # Apply custom rewrite table
          exp-swap-push-push default;
        }
      }
    }
}
```

```
rewrite-rules {
    exp exp_rew {
        forwarding-class be {
            loss-priority low code-point 000;
            loss-priority high code-point 100;
        }
        forwarding-class ef {
            loss-priority low code-point 001;
            loss-priority high code-point 101;
        }
        forwarding-class af {
            loss-priority low code-point 010;
            loss-priority high code-point 110;
        }
        forwarding-class nc {
            loss-priority low code-point 011;
            loss-priority high code-point 111;
        }
    }
}
```

Rewriting IEEE 802.1p Packet Headers with MPLS EXP Value

For Ethernet interfaces installed on a M320 and T-series platform with a peer connection to an M-series routing platform or a T-series platform, you can rewrite both MPLS EXP and IEEE 802.1p bits to a configured value. This allows you to pass the configured value to the Layer 2 VLAN path.

To rewrite both the MPLS EXP and IEEE 802.1p bits, you must include EXP and IEEE 802.1p rewrite rules in the interface configuration. To configure EXP and IEEE 802.1p rewrite rules, include the rewrite-rules statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level, specifying the exp and ieee-802.1 options:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
rewrite-rules {
    exp rewrite-rule-name;
    ieee-802.1 default;
}
```

When you combine these two rewrite rules, only the EXP rewrite table is used for rewriting packet headers. If you do not configure a VLAN on the interface, only the EXP rewriting is in effect. If you do not configure an LSP on the interface or if the MPLS EXP rewrite rule mapping is removed, the IEEE 802.1p default rewrite rules mapping takes effect.

NOTE: You can also combine other rewrite rules. IP, DSCP, DSCP IPv6, and MPLS EXP are associated with Layer 3 packet headers, and IEEE 802.1p is associated with Layer 2 packet headers.

If you combine IEEE 802.1p with IP rewrite rules, the Layer 3 packets and Layer 2 headers are rewritten with the IP rewrite rule.

If you combine IEEE 802.1p with DSCP or DSCP IPv6 rewrite rules, three bits of the Layer 2 header and six bits of the Layer 3 packet header are rewritten with the DSCP or DSCP IPv6 rewrite rule.

The following example shows how to configure an EXP rewrite rule and apply it to both MPLS EXP and IEEE 802.1p bits:

```
[edit class-of-service]
rewrite-rules {
    exp exp-ieee-table {
        forwarding-class best-effort {
            loss-priority low code-point 000;
            loss-priority high code-point 001;
        }
        forwarding-class assured-forwarding {
            loss-priority low code-point 010;
            loss-priority high code-point 011;
        }
        forwarding-class expedited-forwarding {
            loss-priority low code-point 111;
            loss-priority high code-point 110;
        }
        forwarding-class network-control {
            loss-priority low code-point 100;
            loss-priority high code-point 101;
        }
    }
}
interfaces {
    so-3/1/0 {
        unit 0 {
            rewrite-rules {
                exp exp-ieee-table;
                ieee-802.1 default;
            }
        }
    }
}
```

Rewriting Frame Relay Headers

For J-series Services Router interfaces with Frame Relay encapsulation, you can rewrite the discard eligibility (DE) bit based on the loss priority of Frame Relay traffic. For each outgoing frame with the loss priority set to low, medium-low, medium-high, or high, you can set the DE bit CoS value to 0 or 1.

You can combine a Frame Relay rewrite rule with other rewrite rules on the same interface. For example, you can rewrite both the DE bit and MPLS EXP bit.

This section is organized as follows:

Assigning the Default Frame Relay Rewrite Rule to an Interface

The default Frame Relay rewrite rule contains the following settings:

```
loss-priority low code-point 0;
loss-priority medium-low code-point 0;
loss-priority medium-high code-point 1;
loss-priority high code-point 1;
```

This default rule sets the DE CoS value to 0 for each outgoing frame with the loss priority set to low or medium-low. This default rule sets the DE CoS value to 1 for each outgoing frame with the loss priority set to medium-high or high.

To assign the default rule to an interface, include the frame-relay-de default statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
frame-relay-de default;
```

Defining a Custom Frame Relay Rewrite Rule

To define a custom Frame Relay rewrite rule, include the following statements at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
rewrite-rules {
    frame-relay-de rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (0 | 1);
        }
    }
}
```

A custom rewrite rule sets the DE bit to the 0 or 1 CoS value based on the assigned loss priority of low, medium-low, medium-high, or high for each outgoing frame.

Applying the Rule to a Logical Interface

The rule does not take effect until you apply it to a logical interface. To apply a rule to a logical interface, include the frame-relay-de *map-name* statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
frame-relay-de rewrite-name;
```