

Chapter 9

Rewriting Packet Header Information

As the packet leaves the routing platform, the final CoS action is the application of a rewrite rule. Rewrite rules set the value of the CoS bits within the packet's header. The bit setting is determined by the packet's forwarding class and loss priority setting. Each rule examines the forwarding class and loss priority of the packet about to be transmitted and sets the bits to a value defined in the rule. In effect, the rewrite rule performs the opposite function of the behavior aggregate (BA) classifier used when the packet enters the router.

You always apply rewrite rules to egress interfaces, and you generally apply them on the edge of your network. This allows the downstream router in a neighboring network to classify each packet into the appropriate service group.

To configure class-of-service (CoS) rewrite rules, you can include the following statements at the [edit class-of-service] hierarchy level of the configuration:

```
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);
          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 on page 88
- Configuring Rewrite Rules on page 90
- Assigning the Rewrite-Rules Configuration to the Output Logical Interface on page 90
- Rewriting EXP Bits on a Particular Node on page 91
- Rewriting MPLS and IPv4 Packet Headers on page 92
- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet on page 95
- Rewriting IEEE 802.1p Packet Headers with MPLS EXP Value on page 97
- Rewriting Frame Relay Headers on page 98

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;
inet-precedence default;
```

Table 20 on page 89 shows the default rewrite rule mappings. The default rewrite mappings are based on the default bit definitions of DSCP, DSCP IPv6, EXP, IEEE, and IP code points, as shown in Table 8 on page 30, and the default forwarding classes shown in Table 9 on page 34.

When the software detects packets whose CoS settings match the forwarding class and PLP values in the first two columns, the software maps the header bits of those packets to the code-point aliases in the last column. The code-point aliases in the last column map to the bits shown in Table 8 on page 30.

Table 20: 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 30 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 30 shows that this code-point alias maps to the `000` bits.

To evaluate all the implications of this example, see Table 8 on page 30 and Table 20.

```

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

```

Configuring Rewrite Rules

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 14.

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.

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

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 code points.

In addition, the least significant bit of the code point itself must represent the PLP value. For example, code point 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 21.

Table 21: Default MPLS EXP Rewrite Table

| Forwarding Class | Loss Priority | Code Point |
|----------------------|---------------|------------|
| 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 code point 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 code point 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 IP precedence and MPLS EXP values.
- `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 code point 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 IP precedence and MPLS EXP values.

An alternative to overwriting the default with a rewrite-rules mapping is to configure the default packet header rewrite mappings, as shown in Table 20 on page 89.

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.

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 code point value is written to MPLS and IPv4 headers. For VPN MPLS packets without IPv4 payloads, the code point 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 CoS of an incoming MPLS or non-MPLS packet.

To do this on 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;

```

To do this 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 21 on page 92. 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 90 and “Assigning the Rewrite-Rules Configuration to the Output Logical Interface” on page 90.

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 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 code point 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 on page 99
- Defining a Custom Frame Relay Rewrite Rule on page 99

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 code point to 0 for each outgoing frame with the loss priority set to low or medium-low. This default rule sets the DE code point 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 code point 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;
```

