JUNIPER NETWORKS

Junos® OS

CoS Packet Classification Based on Behavior Aggregates Feature Guide for Routing Devices

Release
14.1

Published: 2014-06-25

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS CoS Packet Classification Based on Behavior Aggregates Feature Guide for Routing Devices
14.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation xi
Documentation and Release Notes xi
Supported Platforms xi
Using the Examples in This Manual xi
Merging a Full Example xii
Merging a Snippet xii
Documentation Conventions xiii
Documentation Feedback xv
Requesting Technical Support xv
Self-Help Online Tools and Resources xv
Opening a Case with JTAC xvi

Part 1 Overview

Chapter 1 **Behavior Aggregate Classifier 3**

BA Classifier Overview 3
Overview of BA Classifier Types 5
Default Behavior Aggregate Classification Overview 6
Understanding DSCP Classification for VPLS 7
BA Classifiers and ToS Translation Tables 7
Classifiers and Rewrite Rules at the Global and Physical Interface Levels Overview 8

Chapter 2 **BA Classifier Default Values 9**

Default CoS Values Overview 9
Default IP Precedence Classifier (ipprec-compatibility) 12
Default IP Precedence Classifier (ipprec-default) 13
Default MPLS EXP Classifier 13
Default DSCP and DSCP IPv6 Classifier 14
Default IEEE 802.1p Classifier 16
Default IEEE 802.1ad Classifier 16

Part 2 Configuration

Chapter 3 **Configuration Tasks for Classifiers 21**

Defining Classifiers 21
Importing a Classifier 22
Defining Code Point Aliases for Bit Patterns 22
Applying Classifiers to Logical Interfaces 24
DSCP Classifier Configuration Examples 28

Chapter 4 Configuration Tasks for BA Classifiers 31
Configuring BA Classifiers for Bridged Ethernet 31
Example: Using a Default Classifier 32
Example: Configuring a Custom DSCP Behavior Aggregate Classifier 39
Example: Configuring DSCP Classification for VPLS 51
Example: DSCP IPv6 Rewrites and Forwarding Class Maps 52
Tunneling and BA Classifiers 53
Configuring Classifiers and Rewrite Rules at the Global and Physical Interface Levels 53
Chapter 5 Configuration Task for DSCP IPv6 Classifiers 57
Applying DSCP IPv6 Classifiers 57
Chapter 6 Configuration Tasks for MPLS EXP Classifiers 59
Applying MPLS EXP Classifiers to Routing Instances 59
Configuring Global Classifiers and Wildcard Routing Instances 60
Examples: Applying MPLS EXP Classifiers to Routing Instances 61
Applying MPLS EXP Classifiers for Explicit-Null Labels 63
Chapter 7 Configuration Task for IEEE 802.1ad Classifiers 65
Configuring and Applying IEEE 802.1ad Classifiers 65
Defining Custom IEEE 802.1ad Maps 65
Applying Custom IEEE 802.1ad Maps 65
Verifying Custom IEEE 802.1ad Map Configuration 66
Chapter 8 Configuration Statements 67
BA Classifier Configuration Hierarchy 68
classifiers (Logical Interface) 69
classifiers (Routing Instance) 70
classifiers (Definition) 71
classifiers (Physical Interface) 72
code-point-aliases 73
code-points 73
dscp (AS PIC Classifiers) 74
dscp (Classifier on Physical Interface) 74
dscp (Class of Service Classifier) 75
dscp (Rewrite Rules) 76
dscp-ipv6 (Class-of-Service) 77
exp 78
forwarding-class (BA Classifiers) 79
ieee-802.1 (Classifier on Physical Interface) 79
ieee-802.1 (Rewrite Rules on Logical Interface) 80
ieee-802.1ad 81
import (Classifiers) 82
inet-precedence 82
inet-precedence (Classifier on Physical Interface) 83
interfaces 84
loss-priority (BA Classifiers) 86
routing-instances (Class of Service) 87
system-defaults 88

unit 89

Part 3 Administration

Chapter 9 Operational Commands 93

show class-of-service code-point-aliases 94
show class-of-service translation-table 96

Part 4 Index

Index 103

List of Figures

Part 1 Overview

Chapter 1 **Behavior Aggregate Classifier 3**

Figure 1: How a Classifier Works 3

Part 2 Configuration

Chapter 4 **Configuration Tasks for BA Classifiers 31**

Figure 2: BA Classifier with Two Queues 35
Figure 3: BA Classifier Scenario 36
Figure 4: BA Classifier with Two Queues 40
Figure 5: Behavior Aggregate Classifier Scenario 42

List of Tables

About the Documentation xi

Table 1: Notice Icons xiii
Table 2: Text and Syntax Conventions xiv

Part 1 Overview

Chapter 2 BA Classifier Default Values 9

Table 3: Default CoS Values 10
Table 4: Default IP Precedence Classifier 12
Table 5: Default IP Precedence (ipprec-default) Classifier 13
Table 6: Default MPLS Classifier 14
Table 7: Default DSCP Classifier 14
Table 8: Default IEEE 802.1p Classifier 16
Table 9: Default IEEE 802.1ad Classifier 17

Part 2 Configuration

Chapter 3 Configuration Tasks for Classifiers 21

Table 10: Logical Interface Classifier Combinations 25

Chapter 4 Configuration Tasks for BA Classifiers 31

Table 11: Sample BA Classifier Loss Priority Assignments 42

Chapter 6 Configuration Tasks for MPLS EXP Classifiers 59

Table 12: Default MPLS EXP Classification Table 59

Part 3 Administration

Chapter 9 Operational Commands 93

Table 13: show class-of-service code-point-aliases Output Fields 94
Table 14: show class-of-service translation-table Output Fields 96

About the Documentation

- Documentation and Release Notes on page xi
- Supported Platforms on page xi
- Using the Examples in This Manual on page xi
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xv

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- T Series
- M Series
- MX Series
- ACX Series
- PTX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming

configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xsl; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** *metric***>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** *community-ids* **]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

- Behavior Aggregate Classifier on page 3
- BA Classifier Default Values on page 9

CHAPTER 1

Behavior Aggregate Classifier

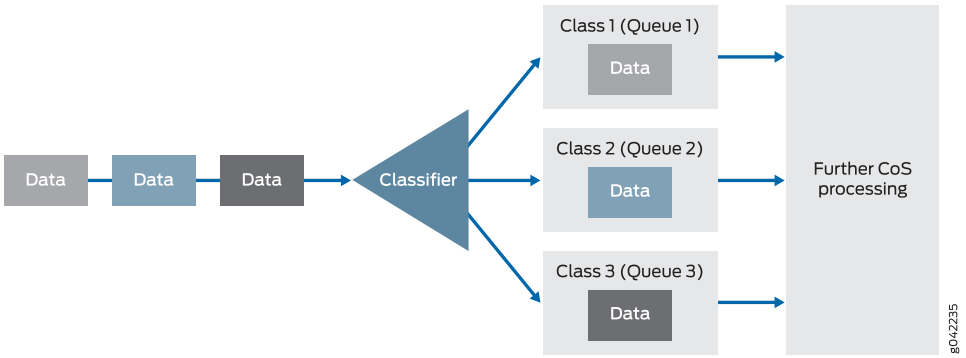
- BA Classifier Overview on page 3
- Overview of BA Classifier Types on page 5
- Default Behavior Aggregate Classification Overview on page 6
- Understanding DSCP Classification for VPLS on page 7
- BA Classifiers and ToS Translation Tables on page 7
- Classifiers and Rewrite Rules at the Global and Physical Interface Levels Overview on page 8

BA Classifier Overview

The behavior aggregate (BA) classifier maps a class-of-service (CoS) value to a forwarding class and loss priority. The forwarding class determines the output queue. The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion.

Figure 1 on page 3 provides a high-level illustration of how a classifier works.

Figure 1: How a Classifier Works



The types of BA classifiers are based on which part of the incoming packet the classifier examines:

- Differentiated Services code point (DSCP) for IP DiffServ
- DSCP for IPv6 DiffServ

- IP precedence bits
- MPLS EXP bits
- IEEE 802.1p CoS bits
- IEEE 802.1ad drop eligible indicator (DEI) bit

Unlike multifield classifiers (which are discussed in *Multifield Classifier Overview*), BA classifiers are based on fixed-length fields, which makes them computationally more efficient than multifield classifiers. For this reason, core devices are normally configured to perform BA classification, because of the higher traffic volumes they handle.

In most cases, you need to rewrite a given marker (IP precedence, DSCP, IEEE 802.1p, IEEE 802.1ad, or MPLS EXP settings) at the ingress node to accommodate BA classification by core and egress devices. For more information about rewrite markers, see *Rewriting Packet Header Information Overview*.

For Juniper Networks M Series Multiservice Edge Routers, four classes can forward traffic independently. For M320 Multiservice Edge Routers and Juniper Networks T Series Core Routers, eight classes can forward traffic independently. Therefore, you must configure additional classes to be aggregated into one of these classes. You use the BA classifier to configure class aggregation.

For Juniper M Series Multiservice Edge Routers with Intelligent Queuing 2 (IQ2) and Enhanced Intelligent Queuing 2 (IQ2E) PICs, the following restrictions apply:

- You can only use BA classifiers for IPv4 DSCP bits for virtual private LAN service (VPLS).
- You cannot use BA classifiers for IPv4 DSCP bits for Layer 2 VPNs.
- You cannot use BA classifiers for IPv6 DSCP bits for VPLS.
- You cannot use BA classifiers for IPv6 DSCP bits for Layer 2 VPNs.

For Juniper Networks MX Series 3D Universal Edge Routers, the following restrictions apply:

- You can only use multifield classifiers (but *not* BA classifiers) for IPv4 DSCP bits for virtual private LAN service (VPLS).
- You cannot use BA classifiers for IPv4 DSCP bits for Layer 2 VPNs.
- You cannot use BA classifiers for IPv6 DSCP bits for VPLS.
- You cannot use BA classifiers for IPv6 DSCP bits for Layer 2 VPNs.

For the 10-port 10-Gigabit Oversubscribed Ethernet (OSE) PICs, the following restrictions on BA classifiers apply:

- Multiple classifiers can be configured to a single logical interface. However, there are some restrictions on which the classifiers can coexist.

  For example, the DSCP and IP precedence classifiers cannot be configured on the same logical interface. The DSCP and IP precedence classifiers can coexist with the DSCP IPv6 classifier on the same logical interface. An IEEE 802.1 classifier can coexist

with other classifiers and is applicable only if a packet does not match any of the configured classifiers. For information about the supported combinations, see "Applying Classifiers to Logical Interfaces" on page 24.

- If the classifiers are not defined explicitly, then the default classifiers are applied as follows:
  - All MPLS packets are classified using the MPLS (EXP) classifier. If there is no explicit MPLS (EXP) classifier, then the default MPLS (EXP) classifier is applied.
  - All IPv4 packets are classified using the IP precedence and DSCP classifiers. If there is no explicit IP precedence or DSCP classifier, then the default IP precedence classifier is applied.
  - All IPv6 packets are classified using a DSCP IPv6 classifier. If there is no explicit DSCP IPv6 classifier, then the default DSCP IPv6 classifier is applied.
  - If the IEEE 802.1p classifier is configured and a packet does not match any explicitly configured classifier, then the IEEE 802.1p classifier is applied.

NOTE: For a specified interface, you can configure both a multifield classifier and a BA classifier without conflicts. Because the classifiers are always applied in sequential order, the BA classifier followed by the multifield classifier, any BA classification result is overridden by a multifield classifier if they conflict. For more information about multifield classifiers, see *Multifield Classifier Overview*.

Overview of BA Classifier Types

The idea behind class of service (CoS) is that packets are not treated identically by the routers or switches on the network. In order to selectively apply service classes to specific packets, the packets of interest must be classified in some fashion.

The simplest way to classify a packet is to use behavior aggregate classification. The DSCP, DSCP IPv6, or IP precedence bits of the IP header convey the behavior aggregate class information. The information might also be found in the MPLS EXP bits, IEEE 802.1ad, or IEEE 802.1p CoS bits.

You can configure the following classifier types:

- DSCP, DSCP IPv6, or IP precedence—IP packet classification (Layer 3 headers)
- MPLS EXP—MPLS packet classification (Layer 2 headers)
- IEEE 802.1p—Packet classification (Layer 2 headers)
- IEEE 802.1ad—Packet classification for IEEE 802.1ad formats (including DEI bit)

If you apply an IEEE 802.1 classifier to a logical interface, this classifier takes precedence and is not compatible with any other classifier type. On Juniper Networks MX Series Ethernet Services Routers and EX Series switches using IEEE 802.1ad frame formats, you can apply classification on the basis of the IEEE 802.1p bits (three bits in either the inner

virtual LAN (VLAN) tag or the outer VLAN tag) and the drop eligible indicator (DEI) bit. On routers with IQ2 PICs using IEEE 802.1ad frame format, you can apply classification based on the IEEE 802.1p bits and the DEI bit. Classifiers for IP (DSCP or IP precedence) and MPLS (EXP) can coexist on a logical interface if the hardware requirements are met. (See "Applying Classifiers to Logical Interfaces" on page 24.)

The Enhanced Queuing DPC (EQ DPC) does not support BA classification for packets received from a Layer 3 routing interface or a virtual routing and forwarding (VRF) interface and routed to an integrated routing and bridging interface (IRB) to reach the remote end of a pseudowire connection. The EQ DPC also does not support BA classification for Layer 2 frames received from a Virtual Private LAN Service (VPLS) pseudowire connection from a remote site and routed to a Layer 3 routing interface through an IRB interface.

Default Behavior Aggregate Classification Overview

The software automatically assigns an implicit default IP precedence classifier to all logical interfaces.

NOTE: **Only the IEEE 802.1p classifier is supported in Layer 2 interfaces. You must explicitly apply this classifier to the interface as shown in** **"Default IEEE 802.1p Classifier" on page 16****.**

If you enable the MPLS protocol family on a logical interface, a default MPLS EXP classifier is automatically applied to that logical interface.

Other default classifiers (such as those for IEEE 802.1p bits and DSCP) require that you explicitly associate a default classification table with a logical interface. When you explicitly associate a default classifier with a logical interface, you are in effect overriding the implicit default classifier with an explicit default classifier.

NOTE: **Although several code points map to the expedited-forwarding (ef) and assured-forwarding (af) classes, by default no resources are assigned to these forwarding classes. All af classes other than af1x are mapped to best-effort, because RFC 2597, *Assured Forwarding PHB Group*, prohibits a node from aggregating classes.**

You can apply IEEE 802.1p classifiers to interfaces that are part of VPLS routing instances.

Related Documentation

- Default IP Precedence Classifier (ipprec-compatibility) on page 12
- Default MPLS EXP Classifier on page 13
- Default DSCP and DSCP IPv6 Classifier on page 14
- Default IEEE 802.1p Classifier on page 16
- Default IEEE 802.1ad Classifier on page 16
- Default IP Precedence Classifier (ipprec-default) on page 13

Understanding DSCP Classification for VPLS

You can perform Differentiated Services Code Point (DSCP) classification for IPv4 packets on Ethernet interfaces that are part of a virtual private LAN service (VPLS) routing instance on the ingress provider edge (PE) router. This is supported on the M320 router with Enhanced type III FPC and the M120 router. On the ATM II IQ PIC, the **ether-vpls-over-atm-llc** encapsulation statement is required. On the Intelligent Queuing 2 (IQ2) or Intelligent Queuing 2 Enhanced (IQ2E) PICs, the **vlan-vpls** encapsulation statement is required. DSCP for IPv6 and Internet precedence for IPv6 are not supported.

In order to perform DSCP classification for IPv4 packets on Ethernet interfaces that are part of a VPLS routing instance on the ingress PE router, you must make sure of the following:

- The correct encapsulation statement based on PIC type is configured for the interface.
- The DSCP classifier is defined (default is allowed) at the **[edit class-of-service classifiers]** hierarchy level.
- The defined DSCP classifier is applied to the interface.
- The interface is included in the VPLS routing instance on the ingress of the PE router.

Related Documentation

- BA Classifier Overview on page 3

BA Classifiers and ToS Translation Tables

On some PICs, the behavior aggregate (BA) translation tables are included for every logical interface (unit) protocol family configured on the logical interface. The proper default translation table is active even if you do not include any explicit translation tables. You can display the current translation table values with the **show class-of-service classifiers** command.

On Juniper Networks M40e, M120, M320 Multiservice Edge Routers, and T Series Core Routers with Enhanced IQ (IQE) PICs, or on any router or switch with IQ2 or Enhanced IQ2 (IQ2E) PICs, you can replace the type-of-service (ToS) bit value on the incoming packet header on a logical interface with a user-defined value. The new ToS value is used for all class-of-service processing and is applied before any other class-of-service or firewall treatment of the packet. The PIC uses the **translation-table** statement to determine the new ToS bit values.

You can configure a physical interface (port) or logical interface (unit) with up to three translation tables. For example, you can configure a port or unit with BA classification for IPv4 DSCP, IPv6 DSCP, and MPLS EXP. The number of frame relay data-link connection identifiers (DLCIs) (units) that you can configure on each PIC varies based on the number and type of BA classification tables configured on the interfaces.

For more information on configuring ToS translation tables, along with examples, see *Configuring ToS Translation Tables*.

Classifiers and Rewrite Rules at the Global and Physical Interface Levels Overview

On ACX Series Universal Access Routers and EX Series switches, CoS supports classification and rewrite at the global level and physical interface levels.

At a global level, you can define EXP classification.

At a physical interface level, you can define the following features:

- DSCP and inet-precedence classifiers
- DSCP and inet-precedence rewrites
- ieee-802.1 classifiers (inner and outer)
- ieee-802.1 rewrites (outer)

At a logical interface level, you can define the fixed classification and EXP rewrites.

To configure global EXP classifiers, include the **classfiers exp *classifier-name*** statement at the **[edit class-of-service] system-defaults** hierarchy level.

To configure classifiers or rewrite rules at the physical interface, include either the **classifiers** statement or the **rewrite-rules** statement at the **[edit class-of-service] interfaces *interface-name*]** hierarchy level.

To display classifiers configured under **system-defaults**, enter the **show class-of-service system-defaults** command.

To display classifiers and rewrite rules bound to physical interfaces, enter the **show class-of-service interfaces *interface-name*** command.

Related Documentation

- Configuring Classifiers and Rewrite Rules at the Global and Physical Interface Levels on page 53

CHAPTER 2

BA Classifier Default Values

- Default CoS Values Overview on page 9
- Default IP Precedence Classifier (ipprec-compatibility) on page 12
- Default IP Precedence Classifier (ipprec-default) on page 13
- Default MPLS EXP Classifier on page 13
- Default DSCP and DSCP IPv6 Classifier on page 14
- Default IEEE 802.1p Classifier on page 16
- Default IEEE 802.1ad Classifier on page 16

Default CoS Values Overview

Behavior aggregate (BA) classifiers use class-of-service (CoS) values—such as Differentiated Services code points (DSCPs), DSCP IPv6, IP precedence, IEEE 802.1, and MPLS experimental (EXP) bits—to associate incoming packets with a particular CoS servicing level. You can assign a meaningful name or alias to the CoS values and use this alias instead of bits when configuring CoS components. These aliases are not part of the specifications but are well known through usage. For example, the alias for DSCP 101110 is widely accepted as **ef** (expedited forwarding).

NOTE: **The code point aliases must begin with a letter and can be up to 64 characters long.**

When you configure classes and define classifiers, you can refer to the markers by alias names. You can configure user-defined classifiers in terms of alias names. If the value of an alias changes, it alters the behavior of any classifier that references it.

To configure class-of-service (CoS) code point aliases, include the **code-point-aliases** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
code-point-aliases {
   (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
      alias-name bits;
   }
}
```

Table 3 on page 10 shows the default mappings between the bit values and standard aliases. For example, it is widely accepted that the alias for DSCP 101110 is **ef** (expedited forwarding).

Table 3: Default CoS Values

CoS Value Types	Mapping
DSCP and DSCP IPv6 CoS Values	
ef	101110
af11	001010
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010
af32	011100
af33	011110
af41	100010
af42	100100
af43	100110
be	000000
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
nc1/cs6	110000
nc2/cs7	111000

Table 3: Default CoS Values *(continued)*

CoS Value Types	Mapping
MPLS EXP CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111
IEEE 802.1 CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101
nc1/cs6	110
nc2/cs7	111
Legacy IP Precedence CoS Values	
be	000
be1	001
ef	010
ef1	011
af11	100
af12	101

Table 3: Default CoS Values *(continued)*

CoS Value Types	Mapping
nc1/cs6	110
nc2/cs7	111

Related Documentation

- Default IP Precedence Classifier (ipprec-compatibility) on page 12
- Default IP Precedence Classifier (ipprec-default) on page 13
- Default MPLS EXP Classifier on page 13
- Default DSCP and DSCP IPv6 Classifier on page 14
- Default IEEE 802.1p Classifier on page 16
- Default IEEE 802.1ad Classifier on page 16
- **code-point-aliases on page 73**

Default IP Precedence Classifier (ipprec-compatibility)

By default, all logical interfaces are automatically assigned an implicit IP precedence classifier called **ipprec-compatibility**. The **ipprec-compatibility** IP precedence classifier maps IP precedence bits to forwarding classes and loss priorities, as shown in Table 4 on page 12.

Table 4: Default IP Precedence Classifier

IP Precedence CoS Values	Forwarding Class	Loss Priority
000	**best-effort**	**low**
001	**best-effort**	**high**
010	**best-effort**	**low**
011	**best-effort**	**high**
100	**best-effort**	**low**
101	**best-effort**	**high**
110	**network-control**	**low**
111	**network-control**	**high**

Default IP Precedence Classifier (ipprec-default)

There are two separate tables for default IP precedence classification. All logical interfaces are implicitly assigned the **ipprec-compatibility** classifier by default, as described in Table 4 on page 12.

The other default IP precedence classifier (called **ipprec-default**) overrides the **ipprec-compatibility** classifier when you explicitly associate it with a logical interface. To do this, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers inet-precedence]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers
  inet-precedence]
default;
```

Table 5 on page 13 shows the forwarding class and PLP that are assigned to the IP precedence CoS bits when you apply the default IP precedence classifier.

Table 5: Default IP Precedence (ipprec-default) Classifier

Code Point	Forwarding Class	PLP
000	**best-effort**	**low**
001	**assured-forwarding**	**low**
010	**best-effort**	**low**
011	**best-effort**	**low**
100	**best-effort**	**low**
101	**expedited-forwarding**	**low**
110	**network-control**	**low**
111	**network-control**	**high**

Default MPLS EXP Classifier

For all PICs except PICs mounted on Juniper Networks M Series Multiservice Edge Router standard (nonenhanced) FPCs, if you enable the MPLS protocol family on a logical interface, the default MPLS EXP classifier is automatically applied to that logical interface.

To configure code point aliases for MPLS EXP CoS markers, map alias names to bit patterns at the **[edit class-of-service code-point-aliases exp]** hierarchy level.

Table 6 on page 14 lists the default MPLS classifier mapping of EXP bits to forwarding classes and loss priorities.

Table 6: Default MPLS Classifier

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

Related Documentation

- Default CoS Values Overview on page 9
- code-point-aliases on page 73

Default DSCP and DSCP IPv6 Classifier

To enable the default DiffServ code point (DSCP) classifier, include the **default** statement at the **[edit class-of-service interfaces** *interface-name* **unit** *unit-number* **classifiers dscp]** hierarchy level.

To enable the default DSCP IPv6 classifier, include the **default** statement at the **[edit class-of-service interfaces** *interface-name* **unit** *unit-number* **classifiers dscp-ipv6]** hierarchy level.

NOTE: If you deactivate or delete the dscp-ipv6 statement from the configuration, the default IPv6 classifier is not activated on the M5, M10, M7i, M10i, M20, M40, M40e, and M160 routing platforms. As a workaround, explicitly specify the default option to the dscp-ipv6 statement.

Table 7 on page 14 shows the forwarding class and packet loss priority (PLP) that are assigned to each well-known DSCP when you apply the explicit default DSCP or DSCP IPv6 classifier.

Table 7: Default DSCP Classifier

DSCP and DSCP IPv6	Forwarding Class	PLP
ef	expedited-forwarding	low

Table 7: Default DSCP Classifier *(continued)*

DSCP and DSCP IPv6	Forwarding Class	PLP
af11	assured-forwarding	low
af12	assured-forwarding	high
af13	assured-forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low
cs3	best-effort	low
cs4	best-effort	low
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network-control	low
other	best-effort	low

Related Documentation

- Overview of BA Classifier Types on page 5
- Default CoS Values Overview on page 9
- *Changing the Default Queuing and Marking of Host Outbound Traffic*

- classifiers (Logical Interface) on page 69

Default IEEE 802.1p Classifier

Table 8 on page 16 shows the forwarding class and PLP that are assigned to the IEEE 802.1p CoS bits when you apply the explicit default IEEE 802.1p classifier. To do this, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers ieee-802.1]** hierarchy level:

NOTE: Only the IEEE 802.1p classifier is supported in Layer 2 interfaces. You must explicitly apply this classifier as shown.

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers
   ieee-802.1]
default;
```

Table 8: Default IEEE 802.1p Classifier

Code Point	Forwarding Class	PLP
000	best-effort	low
001	best-effort	high
010	expedited-forwarding	low
011	expedited-forwarding	high
100	assured-forwarding	low
101	assured-forwarding	high
110	network-control	low
111	network-control	high

Default IEEE 802.1ad Classifier

Table 9 on page 17 shows the code point, forwarding class alias, and PLP that are assigned to the IEEE 802.1ad bits when you apply the explicit default IEEE 802.1ad classifier. The table is very similar to the IEEE 802.1p default table, but the loss priority is determined by the DEI bit. To apply the default table, include the **default** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers ieee-802.1]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers
   ieee-802.1ad]
default;
```

Table 9: Default IEEE 802.1ad Classifier

IEEE 802.1ad Code Point	Forwarding Class Alias	PLP
0000	be	low
0010	be1	low
0100	ef	low
0110	ef1	low
1000	af11	low
1010	af12	low
1100	nc1	low
1110	nc2	low
0001	be-dei	high
0011	be1-dei	high
0101	ef-dei	high
0111	ef1-dei	high
1001	af11-dei	high
1011	af12-dei	high
1101	nc1-dei	high
1111	nc2-dei	high

PART 2

Configuration

- Configuration Tasks for Classifiers on page 21
- Configuration Tasks for BA Classifiers on page 31
- Configuration Task for DSCP IPv6 Classifiers on page 57
- Configuration Tasks for MPLS EXP Classifiers on page 59
- Configuration Task for IEEE 802.1ad Classifiers on page 65
- Configuration Statements on page 67

CHAPTER 3

Configuration Tasks for Classifiers

- Defining Classifiers on page 21
- Defining Code Point Aliases for Bit Patterns on page 22
- Applying Classifiers to Logical Interfaces on page 24
- DSCP Classifier Configuration Examples on page 28

Defining Classifiers

You can override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the **classifiers** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
classifiers {
   (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
      import [classifier-name | default];
      forwarding-class class-name {
         loss-priority level code-points [ aliases ] [ bit-patterns ];
      }
   }
}
```

The map sets the forwarding class and PLP for a specific set of code-point aliases and bit patterns. The inputs of the map are code-point aliases and bit patterns. The outputs of the map are the forwarding class and the PLP. For more information about how CoS maps work, see *CoS Inputs and Outputs Overview*.

The classifiers work as follows:

- **dscp**—Handles incoming IPv4 packets.
- **dscp-ipv6**—Handles incoming IPv6 packets. For more information, see "Applying DSCP IPv6 Classifiers" on page 57.
- **exp**—Handles MPLS packets using Layer 2 headers.
- **ieee-802.1**—Handles Layer 2 CoS.
- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

A classifier takes a specified bit pattern as either the literal pattern or as a defined alias and attempts to match it to the type of packet arriving on the interface. If the information in the packet's header matches the specified pattern, the packet is sent to the appropriate queue, defined by the forwarding class associated with the classifier.

The code-point aliases and bit patterns are the input for the map. The loss priority and forwarding class are outputs of the map. In other words, the map sets the PLP and forwarding class for a given set of code-point aliases and bit patterns.

NOTE: On M Series, MX Series, and T Series routers, and EX Series switches that do not have tricolor marking enabled, the loss priority can be configured only by setting the PLP within a multifield classifier. This setting can then be used by the appropriate drop profile map and rewrite rule. For more information, see *Setting Packet Loss Priority*.

Importing a Classifier

You can use any table, including the default, in the definition of a new classifier by including the **import** statement. The imported classifier is used as a template and is not modified. Whenever you commit a configuration that assigns a new ***class-name*** and **loss-priority** value to a code-point alias or set of bits, it replaces that entry in the imported classifier template. As a result, you must explicitly specify every CoS value in every designation that requires modification.

To do this, include the **import default** statement at the **[edit class-of-service classifiers *type classifier-name*]** hierarchy level:

```
[edit class-of-service classifiers type classifier-name]
import default;
```

For instance, to import the default DSCP classifier, include the **dscp default** statement at the **[edit class-of-service classifiers dscp *classifier-name*]** hierarchy level:

```
[edit class-of-service classifiers dscp classifier-name]
import default;
```

Defining Code Point Aliases for Bit Patterns

To define a code-point alias, include the **code-point-aliases** statement at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
code-point-aliases {
   (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
      alias-name bit-pattern;
   }
}
```

The CoS marker types are as follows:

- **dscp**—Handles incoming IPv4 packets.
- **dscp-ipv6**—Handles incoming IPv6 packets.
- **exp**—Handles MPLS packets using Layer 2 headers.
- **ieee-802.1**—Handles Layer 2 CoS.
- **inet-precedence**—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

For example, you might configure the following aliases:

```
[edit class-of-service]
code-point-aliases {
   dscp {
      my1 110001;
      my2 101110;
      be 000001;
      cs7 110000;
   }
}
```

This configuration produces the following mapping:

```
user@host> show class-of-service code-point-aliases dscp
Code point type: dscp

Alias              Bit pattern

ef/my2             101110

af11               001010

af12               001100

af13               001110

af21               010010

af22               010100

af23               010110

af31               011010

af32               011100

af33               011110

af41               100010

af42               100100

af43               100110

be                 000001

cs1                001000

cs2                010000

cs3                011000

cs4                100000

cs5                101000

nc1/cs6/cs7        110000
```

```
nc2                  111000
my1                  110001
```

The following notes explain certain results in the mapping:

- **my1 110001**:
  - 110001 was not mapped to anything before, and **my1** is a new alias.
  - Nothing in the default mapping table is changed by this statement.
- **my2 101110**:
  - 101110 is now mapped to **my2** as well as **ef**.
- **be 000001**:
  - **be** is now mapped to 000001.
  - The old value of **be**, 000000, is not associated with any alias. Packets with this DSCP value are now mapped to the default forwarding class.
- **cs7 110000**:
  - **cs7** is now mapped to 110000, as well as **nc1** and **cs6**.
  - The old value of **cs7**, 111000, is still mapped to **nc2**.

Related Documentation

- Applying DSCP IPv6 Classifiers on page 57

Applying Classifiers to Logical Interfaces

To apply the classification map to a logical interface:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
user@host#set classifiers (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence)
  (classifier-name | default);
```

You can use interface wildcards for ***interface-name*** and ***logical-unit-number***.

For most PICs, if you apply an IEEE 802.1p classifier to a logical interface, you cannot apply non-IEEE classifiers to other logical interfaces on the same physical interface. This restriction does not apply to Gigabit Ethernet IQ2 PICs.

There are some restrictions on applying multiple BA classifiers to a single logical interface. Table 10 on page 25 shows the supported combinations. In this table, the OSE PICs refer to the 10-port 10-Gigabit OSE PICs.

Table 10: Logical Interface Classifier Combinations

Classifier Combinations	Gigabit Ethernet IQ2 PICs	OSE PICs	Other PICs on M320, MX Series, T Series routers and on EX Series Switches	Other M Series with Regular FPCs	Other M Series with Enhanced FPCs
dscp and **inet-precedence**	No	No	No	No	No
dscp-ipv6 and **(dscp | inet-precedence)**	Yes	Yes	Yes	No	No
exp and **ieee 802.1**	Yes	Yes	No	No	No
ieee 802.1 and **(dscp | dscp-ipv6 | exp | inet-precedence)**	Yes	Yes	No	No	Yes
exp and **(dscp | dscp-ipv6 | inet-precedence)**	Yes	Yes	Yes	No	Yes

For Gigabit Ethernet IQ2 and 10-port 10-Gigabit Oversubscribed Ethernet (OSE) interfaces, family-specific classifiers take precedence over IEEE 802.1p BA classifiers. For example, if you configure a logical interface to use both an MPLS EXP and an IEEE 802.1p classifier, the EXP classifier takes precedence. MPLS-labeled packets are evaluated by the EXP classifier, and all other packets are evaluated by the IEEE 802.1p classifier. The same is true about other classifiers when combined with IEEE 802.1p classifiers on the same logical interface.

In Junos OS Releases 9.6 and later, the DSCP and IPv6 DSCP classifiers are not compatible with older formats. You cannot directly replace the old classifier with the new one. You must first delete the old classifier and then apply the new one, although both steps can be done in one configuration session. Otherwise, the commit will fail.

NOTE: If an interface is mounted on an M Series router FPC, you can apply only the default exp classifier. If an interface is mounted on an enhanced FPC, you can create a new exp classifier and apply it to an interface.

On MX960, MX480, MX240, MX80, M120, and M320 routers and EX Series switches with Enhanced Type III FPCs only, you can configure user-defined DSCP-based BA classification for MPLS interfaces (this feature is not available for IQE PICs or on MX Series routers and EX Series switches when ingress queuing is used) or VPLS/L3VPN routing instances (LSI interfaces). The DSCP-based classification for MPLS packets for Layer 2 VPNs is not supported. To classify MPLS packets on the routing instance at the egress PE, include the **dscp** or **dscp-ipv6** statements at the [**edit class-of-service routing-instances *routing-instance-name* classifiers**] hierarchy level. To classify MPLS packets at the core-facing interface, apply the classifier at the [**edit class-of-service interface**

interface-name unit *unit-name* classifiers (dscp | dscp-ipv6) *classifier-name* family mpls] hierarchy level.

NOTE: If you do not apply a DSCP classifier, the default EXP classifier is applied to MPLS traffic.

You can apply DSCP classification for MPLS traffic in the following usage scenarios:

- In a Layer 3 VPN (L3VPN) using an LSI routing instance.
  - Supported on the M120, M320, MX960, MX480, MX240, and MX80 routers.
  - DSCP classifier configured under **[edit class-of-service routing-instances]** on the egress PE router.
- In VPLS using an LSI routing instance.
  - Supported on the M120, M320, MX960, MX480, MX240, and MX80 routers.
  - DSCP classifier configured under **[edit class-of-service routing-instances]** on the egress PE router.
- In a Layer 3 VPN (L3VPN) using a VT routing instance.
  - Supported on the M120, M320, MX960, MX480, MX240, and MX80 routers.
  - DSCP classifier configured under **[edit class-of-service interfaces]** on the core-facing interface on the egress PE router.
- In VPLS using the VT routing instance.
- MPLS forwarding.
  - Supported on the M120, M320, MX960, MX480, MX240, and MX80 routers (not supported on IQE and MX when ingress queuing is enabled).
  - DSCP classifier configured under **[edit class-of-service interfaces]** on the ingress core-facing interface on the P or egress PE router.

MPLS forwarding when the label stacking is greater than 2 is not supported:

The following example configures a DSCP classifier for IPv4 named **dscp-ipv4-classifier** for the **fc-af11-class** forwarding class and a corresponding IPv6 DSCP classifier:

```
class-of-service {
  routing-instances routing-instance-one {
    classifiers {
      dscp dscp-ipv4-classifier {
        loss-priority low code-points 000100;
      }
      dscp dscp-ipv6-classifier {
        forwarding-class fc-af11-class {
          loss-priority low {
            code-points af11;
          }
        }
```

```
        }
      }
    }
  }
```

NOTE: This is not a complete configuration.

This example applies the IPv4 classifier to MPLS traffic and the IPv6 classifier to Internet traffic on interface **ge-2/0/3.0**:

```
class-of-service {
  interfaces ge-2/0/3 {
    unit 0 {
      classifiers {
        dscp dscp-ipv4-classifier {
          family mpls;
        }
        dscp-ipv6 dscp-ipv6-classifier {
          family inet; # This is the default if not present.
        }
      }
    }
  }
}
```

NOTE: This is not a complete configuration.

This example applies the same classifier to both MPLS and IP traffic on interface **ge-2/2/0**.

```
[edit class-of-services interface ge-2/2/0]
unit 0 {
  classifiers {
    dscp dscp-mpls {
      family [ mpls inet ];
    }
  }
}
```

NOTE: This is not a complete configuration.

NOTE: You can apply DSCP and DSCP IPv6 classifiers to explicit null MPLS packets. The **family mpls** statement works the same on both explicit null and non-null MPLS labels.

Related Documentation

- DSCP Classifier Configuration Examples on page 28

DSCP Classifier Configuration Examples

On MX960, MX480, MX240, MX80, M120, and M320 routers with Enhanced Type III FPCs and EX Series switches only, you can configure user-defined DSCP-based BA classification for MPLS interfaces (this feature is not available for IQE PICs or on MX Series routers or EX Series switches when ingress queuing is used) or VPLS/L3VPN routing instances (LSI interfaces).The following examples show how you can apply DSCP classifiers for MPLS traffic in these cases.

Applying a DSCP Classifier to MPLS Packets on the Core-facing Interface

Configure the core-facing interface and associated logical interfaces:

```
interfaces ge-5/3/1 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
    family iso;
    family inet6 {
      address 2000::1/64;
    }
    family mpls
  }
```

Configure the DSCP classifier.

```
class-of-service {
  classifiers {
    dscp dscp11 {
      forwarding-class expedited-forwarding {
        loss-priority low code-points [ ef cs5 ];
      }
      forwarding-class assured-forwarding {
        loss-priority low code-points [ af21 af31 af41 cs4 ];
        loss-priority high code-points [ af23 af33 af43 cs2 af22 af32 af42 cs3 ];
      }
      forwarding-class best-effort {
        loss-priority low code-points [ af11 cs1 af12 ];
        loss-priority high code-points af13;
      }
      forwarding-class network-control {
        loss-priority low code-points [ cs6 cs7 ];
      }
    }
  }
}
```

Attach the classifier to the logical interface for the mpls family. You cannot configure more than one classifier per family.

```
class-of-service {
  interfaces {
    ge-5/3/1 {
      unit 0 {
        classifiers {
          dscp dscp11 {
```

```
                family mpls;
              }
            }
          }
        }
      }
    }
```

The above classifiers are applicable on egress PE routers for VPLS and L3VPN cases. For plain interfaces (not VPLS/L3VPN (LSI) interfaces), these classifiers are applicable on P and egress PE routers on core facing interfaces.

Applying a DSCP Classifier to MPLS Traffic for L3VPN/VPLS

Configure routing instances of type either vrf or vpls.

```
routing-instances {
  vpls1 {
    instance-type vpls;
    interface ge-2/2/2.0; #customer facing interface for VPLS
    route-distinguisher 10.255.245.51:1;
    vrf-target target:1234:1;
    protocols {
      vpls {
        site-range 10;
        no-tunnel-services;
        site vpls-1-site-1 {
          site-identifier 1;
        }
      }
    }
  }
}
```

Configure the DSCP classifier.

```
class-of-service {
  classifiers {
    dscp dscp11 {
      forwarding-class expedited-forwarding {
        loss-priority low code-points [ ef cs5 ];
      }
      forwarding-class assured-forwarding {
        loss-priority low code-points [ af21 af31 af41 cs4 ];
        loss-priority high code-points [ af23 af33 af43 cs2 af22 af32 af42 cs3 ];
      }
      forwarding-class best-effort {
        loss-priority low code-points [ af11 cs1 af12 ];
        loss-priority high code-points af13;
      }
      forwarding-class network-control {
        loss-priority low code-points [ cs6 cs7 ];
      }
    }
  }
}
```

Attach the classifier to a routing instance. You cannot configure more than one classifier per routing instance.

```
class-of-service {
  routing-instances {
    vpls1 {
      classifiers {
        dscp dscp11;
      }
    }
  }
}
```

Related Documentation

- Applying Classifiers to Logical Interfaces on page 24

CHAPTER 4

Configuration Tasks for BA Classifiers

- Configuring BA Classifiers for Bridged Ethernet on page 31
- Example: Using a Default Classifier on page 32
- Example: Configuring a Custom DSCP Behavior Aggregate Classifier on page 39
- Example: Configuring DSCP Classification for VPLS on page 51
- Example: DSCP IPv6 Rewrites and Forwarding Class Maps on page 52
- Tunneling and BA Classifiers on page 53
- Configuring Classifiers and Rewrite Rules at the Global and Physical Interface Levels on page 53

Configuring BA Classifiers for Bridged Ethernet

On M120 and M320 routers equipped with IQ2 PICs, you can configure BA classification based on the IEEE 802.1 bits for bridged Ethernet over Asynchronous Transfer Mode (ATM), Point-to-Point Protocol (PPP), and frame relay for VPLS applications. The BA classification is applied to the first (outer) tag when tagged frames are received. Untagged frames are bypassed and a value of 000 for the classification IEEE 802.1p bits is assumed. There is no support for circuit cross-connect (CCC), and only port-mode VPLS is supported (in port-mode VPLS, only VLANs on a single physical port are included in the VPLS instance). There is no support for multilink PPP bonding with VPLS. For bridging over frame relay, only frames that do not preserve the frame check sequence (FCS) field are supported. Frames that preserve the FCS field are silently discarded.

The bridging over PPP function is restricted:

- There is no support for “tinygram” compression and expansion.
- Frames received with preserved FCS bits are silently discarded.
- Bridge control frames are also classified based on header bit values.
- Both tagged and untagged frames are classified and forwarded. The peer must discard frame types that are not supported.

This example applies an IEEE 802.1p classifier named **ppp-ether-vpls-classifier** to an interface (**so-1/2/3**) with Ethernet VPLS over PPP encapsulation. Note that the interface and CoS configuration must be consistent to support the feature. You must also configure the classifier and other CoS parameters such as forwarding classes.

```
[edit class-of-service]
interfaces {
   so-1/2/3 {
      unit 0 {
         classifiers {
            ieee-802.1 ppp-ether-vpls-classifier;
         }
      }
   }
}

[edit interfaces]
s0-1/2/3 {
   encapsulation ether-vpls-over-ppp;
   unit 0 {
      family vpls;
   }
}
```

On routers with IQ2 or IQ2E PICs, you can perform BA classification based on the value of the inner VLAN tag in an Ethernet frame. To configure BA classification based on the inner VLAN tag value, include the **inner** option at the [**edit class-of-service interfaces** *interface-name* **unit** *logical-unit-number* **classifiers ieee-802.1** *classifier-name* **vlan-tag**] hierarchy level. You must configure the inner VLAN tag for the logical interface with the ***inner*** option at the [**edit interfaces** *interface-name* **unit** *logical-interface-name* **vlan-tag]** hierarchy level.

```
[edit class-of-service interfaces ge-2/2/2 unit 0]
classifiers ieee-802.1 inner-vlan-tag-ba-classifier {
   vlan-tag inner;
}
```

Example: Using a Default Classifier

A Junos OS classifier identifies and separates traffic flows and provides the means to prioritize traffic later in the class-of-service (CoS) process.

A behavior aggregate (BA) classifier performs this function by associating well-known code points with forwarding classes and loss priorities. To enable a default classifier, you simply apply it to your device interfaces. If a default classifier is not applied to an interface, it does not take effect.

Junos OS provides multiple default BA classifier types, which you can combine and supplement with custom BA classifiers as needed to achieve your overall traffic

classification goals. This example shows how to apply the default (BA) DiffServ code point (DSCP) classifier and verify its functionality.

- Requirements on page 33
- Overview on page 33
- Configuration on page 36
- Verification on page 38

Requirements

To verify this procedure, this example uses a traffic generator. The traffic generator can be hardware-based or it can be software running on a server or host machine. If you do not have access to a traffic generator, you can use extended ping for verification. This approach is shown as well.

The functionality in this procedure is widely supported on devices that run Junos OS. The example shown here was tested and verified on MX Series routers running Junos OS Release 10.4.

Overview

The basis of Junos OS CoS is traffic differentiation. Assigning traffic to different classes of service provides the necessary differentiation. From the point of view of a router, the class of service assigned to a packet defines how the router behaves toward the packet. The concept of traffic differentiation is present in every CoS tool, and as a result, classes of service are present across the entire CoS design. A classifier has one input, the incoming packet, and it has *N* possible outputs, where *N* is the number of possible classes of service into which the packet can be classified.

BA classification is used when the traffic coming into your device already has CoS markings in the packet header. For example, the default DSCP BA classifier specifies that packets coming in with code points 000000 are assigned to the best-effort forwarding class and given a loss priority of low.

A forwarding class and loss priority are assigned by default to each well-known DSCP. To view this, run the **show class-of-service classifier** command.

```
user@host> show class-of-service classifier type dscp

Classifier: dscp-default, Code point type: dscp, Index: 7
  Code point         Forwarding class                  Loss priority
  000000             best-effort                       low
  000001             best-effort                       low
  000010             best-effort                       low
  000011             best-effort                       low
  000100             best-effort                       low
  000101             best-effort                       low
  000110             best-effort                       low
  000111             best-effort                       low
  001000             best-effort                       low
  001001             best-effort                       low
  001010             assured-forwarding                low
  001011             best-effort                       low
  001100             assured-forwarding                high
```

```
001101              best-effort                    low
001110              assured-forwarding             high
001111              best-effort                    low
010000              best-effort                    low
010001              best-effort                    low
010010              best-effort                    low
010011              best-effort                    low
010100              best-effort                    low
010101              best-effort                    low
010110              best-effort                    low
010111              best-effort                    low
011000              best-effort                    low
011001              best-effort                    low
011010              best-effort                    low
011011              best-effort                    low
011100              best-effort                    low
011101              best-effort                    low
011110              best-effort                    low
011111              best-effort                    low
100000              best-effort                    low
100001              best-effort                    low
100010              best-effort                    low
100011              best-effort                    low
100100              best-effort                    low
100101              best-effort                    low
100110              best-effort                    low
100111              best-effort                    low
101000              best-effort                    low
101001              best-effort                    low
101010              best-effort                    low
101011              best-effort                    low
101100              best-effort                    low
101101              best-effort                    low
101110              expedited-forwarding           low
101111              best-effort                    low
110000              network-control                low
110001              best-effort                    low
110010              best-effort                    low
110011              best-effort                    low
110100              best-effort                    low
110101              best-effort                    low
110110              best-effort                    low
110111              best-effort                    low
111000              network-control                low
111001              best-effort                    low
111010              best-effort                    low
111011              best-effort                    low
111100              best-effort                    low
111101              best-effort                    low
111110              best-effort                    low
111111              best-effort                    low
```

The forwarding class determines the output queue. By default, all best-effort traffic uses queue 0.

To view the queues that are associated, by default, with each forwarding class, use the **show class-of-service forwarding-class** command. (For clarity, some of the output is excluded.)

```
user@host> show class-of-service forwarding-class
```

```
Forwarding class                  ID      Queue
  best-effort                      0       0
  expedited-forwarding             1       1
  assured-forwarding               2       2
  network-control                  3       3
```

The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion. When you are thinking about loss priorities, keep in mind that unless you configure them, they have no meaning. The default drop behavior is to wait until the queue is 100 percent full and then begin dropping packets indiscriminately. When the queue dips below 100 percent full, packets stop dropping.

The default drop behavior is shown in the **show class-of-service drop-profile** command.

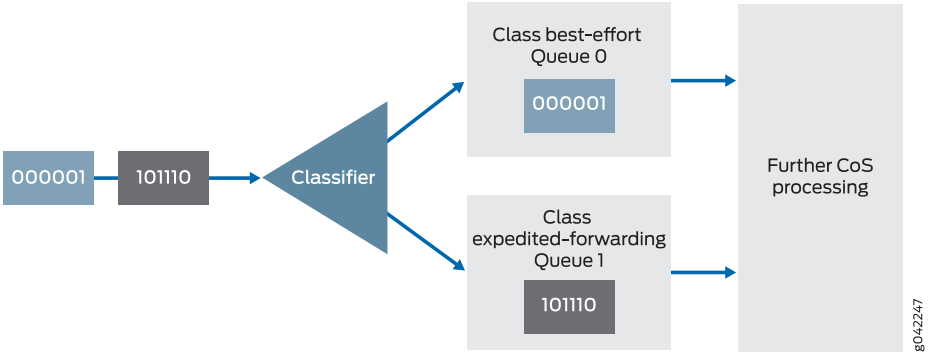
```
user@host> show class-of-service drop-profile

Drop profile: <default-drop-profile>, Type: discrete, Index: 1
  Fill level    Drop probability
         100                 100
```

To create meanings for the various loss priorities, you must configure custom drop profiles. For example, you might define the low loss priority to mean a 10 percent drop probability when the queue is 75 percent full and a 40 percent drop probability when the queue fill level is 95 percent. You might define the high loss priority to mean a 50 percent drop probability when the fill level is 25 percent and a 90 percent drop probability when the fill level is 50 percent. Custom drop profiles are not included in this example, but are mentioned here for clarity because classifiers assign loss priorities. It is important to understand that these assignments are meaningless until you create drop profiles.

The default classifier operation is shown in Figure 2 on page 35. The figure shows two IPv4 packets entering an interface and being classified according to the DSCP code points in the packet headers.

Figure 2: BA Classifier with Two Queues



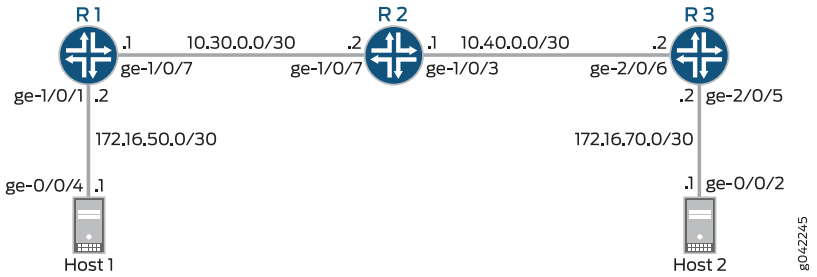
Classifiers are described in more detail in the following Juniper Networks Learning Byte video.

Video: Class of Service Basics, Part 2: Classification Learning Byte

Topology

Figure 3 on page 36 shows the sample network.

Figure 3: BA Classifier Scenario



It is important to apply your class-of-service configuration across the topology, instead of applying it to a single device. Furthermore, even though classification takes effect on incoming interfaces, you should apply BA classifiers to all core and core-facing interfaces. This is because a single interface can be either incoming or outgoing, depending on the direction of the traffic. For example, as traffic flows from Host 1 to Host 2, the incoming interfaces are ge-1/0/9 on Device R2 and ge-2/0/6 on Device R3. As traffic flows in the other direction, from Host 2 to Host R1, the incoming interfaces are ge-1/0/3 on Device R2 and ge-1/0/9 on Device R1.

The BA classifier is not applied to ge-1/0/1 on Device R1 or ge-2/0/5 on Device R3, because these interfaces are not core facing. Generally, at the edge-facing interfaces, you would use a multifield classifier, not a BA classifier.

"CLI Quick Configuration" on page 36 shows the configuration for all of the Juniper Networks devices in Figure 3 on page 36. The section "Step-by-Step Procedure" on page 37 describes the steps on Device R2.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces ge-1/0/1 unit 0 family inet address 172.16.50.2/30
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.1/30
set class-of-service interfaces ge-1/0/9 unit 0 classifiers dscp default
```

Device R2

```
set interfaces ge-1/0/3 unit 0 family inet address 10.40.0.1/30
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.2/30
set class-of-service interfaces ge-1/0/3 unit 0 classifiers dscp default
set class-of-service interfaces ge-1/0/9 unit 0 classifiers dscp default
```

Device R3

```
set interfaces ge-2/0/5 unit 0 family inet address 172.16.70.2/30
set interfaces ge-2/0/6 unit 0 family inet address 10.40.0.2/30
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To enable the default DSCP behavior aggregate classifier:

1. Configure the device interfaces.

```
[edit interfaces]
user@R2# set ge-1/0/3 unit 0 family inet address 10.40.0.1/30
user@R2# set ge-1/0/9 unit 0 family inet address 10.30.0.2/30
```

2. Enable the default DSCP classifier on the interfaces.

```
[edit class-of-service interfaces]
user@R2# set ge-1/0/3 unit 0 classifiers dscp default
user@R2# set ge-1/0/9 unit 0 classifiers dscp default
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** and **show class-of-service** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
ge-1/0/3 {
    unit 0 {
        family inet {
            address 10.40.0.1/30;
        }
    }
}
ge-1/0/9 {
    unit 0 {
        family inet {
            address 10.30.0.2/30;
        }
    }
}
```

```
user@R2# show class-or-service
interfaces {
    ge-1/0/3 {
        unit 0 {
            classifiers {
                dscp default;
            }
        }
    }
    ge-1/0/9 {
        unit 0 {
            classifiers {
                dscp default;
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying Behavior Aggregate Classifiers

Purpose Verify that the default behavior aggregate classifier is enabled on the device interfaces. Keep in mind that although the classifier operates on incoming packets, you view the resulting queue assignment on the outgoing interface.

Action

1. Clear the interface statistics on Device R2.

   ```
   user@R2> clear interface statistics ge-1/0/3
   ```

2. Using extended ping from Device R1 or a packet generator running on a host or server, send packets with the code point set to 101110.

   Both methods are shown here. The packet generator used is hping.

   - When you are using extended ping to set the DSCP code points in the IPv4 packet header, the type-of-service (ToS) decimal value (in this case, 40) is required in the **tos** option of the **ping** command.
   - When you are using hping to set the DSCP code points in the IPv4 packet header, the ToS hex value (in this case, 28) is required in the **--tos** option of the **hping** command.

   If your binary-to-hex or binary-to-decimal conversion skills are rusty, you can use an online calculator, such as http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html .

   NOTE: When you convert a binary DSCP code point value, be sure to add two extra zeros at the end. So instead of 001010, use 00101000. These 0 values (the 7th and 8th bits) are reserved and ignored, but if you do not include them in the conversion, your hex and decimal values will be incorrect.

   ```
   user@R1> ping 172.16.70.1 tos 40 rapid count 25

   PING 172.16.70.1 (172.16.70.1): 56 data bytes
   !!!!!!!!!!!!!!!!!!!!!!!!!
   --- 172.16.70.1 ping statistics ---
   25 packets transmitted, 25 packets received, 0% packet loss
   round-trip min/avg/max/stddev = 0.430/0.477/0.847/0.079 ms

   root@host1> hping 172.16.70.1 --tos 28 -c 25

   HPING 172.16.70.1 (eth1 172.16.70.1): NO FLAGS are set, 40 headers + 0 data
   bytes
   len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.3 ms
   len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.6 ms
   len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=0.4 ms
   len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=0.4 ms
   len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=0.6 ms
   ```

```
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=0.3 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=7 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=20 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=21 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=22 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=23 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=24 win=0 rtt=0.4 ms
```

3. On Device R2, verify that queue 2 is incrementing.

   Code point 101110 is associated with assured-forwarding, which uses queue 2 by default.

   ```
   user@R2> show interfaces extensive ge-1/0/3 | find "queue counters"

   Queue counters:     Queued packets  Transmitted packets      Dropped packets
       0                   0               0                       0
       1                   0               0                       0
       2                   50              25                      0
       3                   3               3                       0
     Queue number:         Mapped forwarding classes
       0                   best-effort
       1                   expedited-forwarding
       2                   assured-forwarding
       3                   network-control
   ```

Meaning The output shows that queue 2 has incremented by 50 packets after sending 50 packets through the router.

Related Documentation

- *Classification Overview*
- *Sample Behavior Aggregate Classification*
- *Understanding Packet Loss Priorities*

Example: Configuring a Custom DSCP Behavior Aggregate Classifier

This example shows how to configure a custom behavior aggregate classifier for a device to determine the forwarding treatment of packets.

Behavior aggregate (BA) classification is used when the traffic coming into your device already has class-of-service (CoS) markings in the packet header. For example, a custom BA classifier might specify that packets coming in with code points set to 101111 be given

a loss priority of low and assigned to the forwarding class Premium-data, which is associated with queue 1. The forwarding class determines the output queue. The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion.

- Requirements on page 40
- Overview on page 40
- Configuration on page 43
- Verification on page 46

Requirements

To verify this procedure, this example uses a traffic generator. The traffic generator can be hardware-based or it can be software running on a server or host machine. If you do not have access to a traffic generator, you can use extended ping for verification. This approach is shown as well.

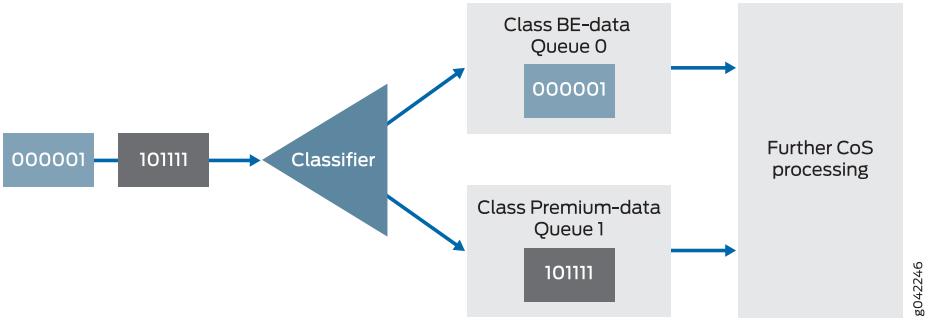
The functionality in this procedure is widely supported on devices that run Junos OS. The example shown here was tested and verified on MX Series routers running Junos OS Release 10.4.

Overview

You configure behavior aggregate classifiers to classify packets that contain valid code points in the packet header. The classifier analyzes the code point in the incoming packet header and assigns an outgoing queue (based on the forwarding class) and a loss priority.

The classifier operation is shown in Figure 2 on page 35.

Figure 4: BA Classifier with Two Queues



Classifiers are described in more detail in the following Juniper Networks Learning Byte video.

Video: Class of Service Basics, Part 2: Classification Learning Byte

Once configured, you must apply the behavior aggregate classifier to the correct interfaces. Generally, core-router interfaces (and core-facing interfaces on edge routers) use behavior aggregate classifiers.

Before you configure a custom classifier, determine the default forwarding class and loss priority for each well-known DiffServ code point (DSCP). To do this, run the **show class-of-service classifier** command. This command lists the code points and the associated forwarding class for the default DSCP classifier, as well as the loss priority defined by default.

```
user@host> show class-of-service classifier type dscp

Classifier: dscp-default, Code point type: dscp, Index: 7
  Code point         Forwarding class                  Loss priority
  000000             best-effort                       low
  000001             best-effort                       low
  000010             best-effort                       low
  000011             best-effort                       low
  000100             best-effort                       low
  000101             best-effort                       low
  000110             best-effort                       low
  000111             best-effort                       low
  001000             best-effort                       low
  001001             best-effort                       low
  001010             assured-forwarding                low
  001011             best-effort                       low
  001100             assured-forwarding                high
  001101             best-effort                       low
  001110             assured-forwarding                high
  001111             best-effort                       low
  010000             best-effort                       low
  010001             best-effort                       low
  010010             best-effort                       low
  010011             best-effort                       low
  010100             best-effort                       low
  010101             best-effort                       low
  010110             best-effort                       low
  010111             best-effort                       low
  011000             best-effort                       low
  011001             best-effort                       low
  011010             best-effort                       low
  011011             best-effort                       low
  011100             best-effort                       low
  011101             best-effort                       low
  011110             best-effort                       low
  011111             best-effort                       low
  100000             best-effort                       low
  100001             best-effort                       low
  100010             best-effort                       low
  100011             best-effort                       low
  100100             best-effort                       low
  100101             best-effort                       low
  100110             best-effort                       low
  100111             best-effort                       low
  101000             best-effort                       low
  101001             best-effort                       low
  101010             best-effort                       low
  101011             best-effort                       low
  101100             best-effort                       low
  101101             best-effort                       low
  101110             expedited-forwarding              low
  101111             best-effort                       low
  110000             network-control                   low
  110001             best-effort                       low
```

```
110010          best-effort             low
110011          best-effort             low
110100          best-effort             low
110101          best-effort             low
110110          best-effort             low
110111          best-effort             low
111000          network-control         low
111001          best-effort             low
111010          best-effort             low
111011          best-effort             low
111100          best-effort             low
111101          best-effort             low
111110          best-effort             low
111111          best-effort             low
```

Applying this information allows you to modify only a few terms and combine them with the default classifier to create a greater scope for classification. When creating a custom classifier, it is useful to include the related default classifier, because then you only have to change a few settings. This example shows how to apply the default classifier within your custom classifier.

Table 11 on page 42 shows how the behavior aggregate classifier used in this example assigns forwarding classes, output queues, and loss priorities based on the code points in the headers of the incoming packets.

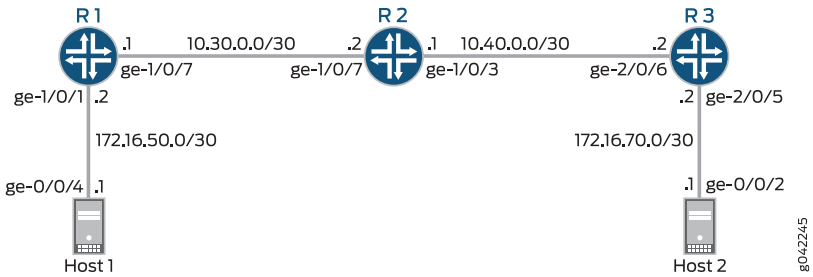
Table 11: Sample BA Classifier Loss Priority Assignments

Incoming Code Points	Outgoing Forwarding Class and Output Queue	Outgoing Loss Priority
000000 (be)	BE-data, queue 0	High
000001 (be1)	BE-data, queue 0	Low
101110 (ef)	Premium-data, queue 1	High
101111 (ef1)	Premium-data, queue 1	Low

Topology

Figure 5 on page 42 shows the sample network.

Figure 5: Behavior Aggregate Classifier Scenario



"CLI Quick Configuration" on page 43 shows the configuration for all of the Juniper Networks devices in Figure 5 on page 42.

The section "Step-by-Step Procedure" on page 44 describes the steps on Device R2.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces ge-1/0/1 unit 0 family inet address 172.16.50.2/30
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.1/30
set class-of-service classifiers dscp v4-ba-classifier import default
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  high code-points be
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  low code-points be1
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority high code-points ef
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority low code-points ef1
set class-of-service code-point-aliases dscp be1 000001
set class-of-service code-point-aliases dscp ef1 101111
set class-of-service forwarding-classes class BE-data queue-num 0
set class-of-service forwarding-classes class Premium-data queue-num 1
set class-of-service forwarding-classes class Voice queue-num 2
set class-of-service forwarding-classes class NC queue-num 3
set class-of-service interfaces ge-1/0/9 unit 0 classifiers dscp v4-ba-classifier
```

Device R2

```
set interfaces ge-1/0/3 unit 0 family inet address 10.40.0.1/30
set interfaces ge-1/0/9 unit 0 family inet address 10.30.0.2/30
set class-of-service classifiers dscp v4-ba-classifier import default
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  high code-points be
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  low code-points be1
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority high code-points ef
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority low code-points ef1
set class-of-service code-point-aliases dscp be1 000001
set class-of-service code-point-aliases dscp ef1 101111
set class-of-service forwarding-classes class BE-data queue-num 0
set class-of-service forwarding-classes class Premium-data queue-num 1
set class-of-service forwarding-classes class Voice queue-num 2
set class-of-service forwarding-classes class NC queue-num 3
set class-of-service interfaces ge-1/0/3 unit 0 classifiers dscp v4-ba-classifier
set class-of-service interfaces ge-1/0/9 unit 0 classifiers dscp v4-ba-classifier
```

Device R3

```
set interfaces ge-2/0/5 unit 0 family inet address 172.16.70.2/30
set interfaces ge-2/0/6 unit 0 family inet address 10.40.0.2/30
set class-of-service classifiers dscp v4-ba-classifier import default
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  high code-points be
set class-of-service classifiers dscp v4-ba-classifier forwarding-class BE-data loss-priority
  low code-points be1
```

```
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority high code-points ef
set class-of-service classifiers dscp v4-ba-classifier forwarding-class Premium-data
  loss-priority low code-points ef1
set class-of-service code-point-aliases dscp be1 000001
set class-of-service code-point-aliases dscp ef1 101111
set class-of-service forwarding-classes class BE-data queue-num 0
set class-of-service forwarding-classes class Premium-data queue-num 1
set class-of-service forwarding-classes class Voice queue-num 2
set class-of-service forwarding-classes class NC queue-num 3
set class-of-service interfaces ge-2/0/6 unit 0 classifiers dscp v4-ba-classifier
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a custom behavior aggregate classifier:

1. Configure the device interfaces.

   ```
   [edit interfaces]
   user@R2# set ge-1/0/3 unit 0 family inet address 10.40.0.1/30
   user@R2# set ge-1/0/9 unit 0 family inet address 10.30.0.2/30
   ```

2. (Optional) Import the default classifier into your customer classifier.

   Importing the default classifier of the same CoS type is useful, because doing so limits the amount of customization required.

   ```
   [edit class-of-service classifiers dscp v4-ba-classifier ]
   user@R2# set import default
   ```

3. Associate the forwarding classes with output queues.

   ```
   [edit class-of-service forwarding-classes]
   user@R2# set class BE-data queue-num 0
   user@R2# set class Premium-data queue-num 1
   user@R2# set class Voice queue-num 2
   user@R2# set class NC queue-num 3
   ```

4. (Optional) For the code points that are not included in the default classifier, define code-point aliases.

   Code-point aliases enable you to use memorable names throughout the configuration, rather than using the binary code points.

   ```
   [edit class-of-service code-point-aliases]
   user@R2# set dscp be1 000001
   user@R2# set dscp ef1 101111
   ```

5. Assign forwarding classes and loss priorities to the code-point aliases.

   ```
   [edit class-of-service classifiers dscp v4-ba-classifier]
   user@R2# set forwarding-class BE-data loss-priority high code-points be
   user@R2# set forwarding-class BE-data loss-priority low code-points be1
   user@R2# set forwarding-class Premium-data loss-priority high code-points ef
   user@R2# set forwarding-class Premium-data loss-priority low code-points ef1
   ```

6. Apply the classifier to the core interfaces.

```
[edit class-of-service]
user@R2# set interfaces ge-1/0/3 unit 0 classifiers dscp v4-ba-classifier
user@R2# set interfaces ge-1/0/9 unit 0 classifiers dscp v4-ba-classifier
```

NOTE: For core devices, interface wildcards are useful. For example:

```
[edit class-of-service]
user@R2# set interfaces ge-* unit * classifiers dscp v4-ba-classifier
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show class-of-service** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
ge-1/0/3 {
  unit 0 {
    family inet {
      address 10.40.0.1/30;
    }
  }
}
ge-1/0/9 {
  unit 0 {
    family inet {
      address 10.30.0.2/30;
    }
  }
}
```

```
user@R2# show class-or-service
classifiers {
  dscp v4-ba-classifier {
    import default;
    forwarding-class BE-data {
      loss-priority high code-points be;
      loss-priority low code-points be1;
    }
    forwarding-class Premium-data {
      loss-priority high code-points ef;
      loss-priority low code-points ef1;
    }
  }
}
code-point-aliases {
  dscp {
    be1 000001;
    ef1 101111;
  }
}
forwarding-classes {
  class BE-data queue-num 0;
  class Premium-data queue-num 1;
  class Voice queue-num 2;
```

```
    class NC queue-num 3;
  }
  interfaces {
    ge-1/0/3 {
      unit 0 {
        classifiers {
          dscp v4-ba-classifier;
        }
      }
    }
    ge-1/0/9 {
      unit 0 {
        classifiers {
          dscp v4-ba-classifier;
        }
      }
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying the Code-Point Aliases on page 46
- Verifying the DSCP Classifier on page 47
- Verifying the Forwarding Classes and Output Queues on page 48
- Verifying That the Classifier Is Applied to the Interfaces on page 49
- Verifying Behavior Aggregate Classifiers on page 49

Verifying the Code-Point Aliases

Purpose Make sure that the code-point aliases are configured as expected.

Action On Device R2, run the **show class-of-service code-point-aliases dscp** command.

```
user@R2> show class-of-service code-point-aliases dscp

Code point type: dscp
  Alias               Bit pattern
  af11                001010
  af12                001100
  af13                001110
  af21                010010
  af22                010100
  af23                010110
  af31                011010
  af32                011100
  af33                011110
  af41                100010
  af42                100100
  af43                100110
  be                  000000
```

```
be1             000001
cs1             001000
cs2             010000
cs3             011000
cs4             100000
cs5             101000
cs6             110000
cs7             111000
ef              101110
ef1             101111
nc1             110000
nc2             111000
```

Meaning The code-point aliases are configured as expected. Notice that the custom aliases that you configure are added to the default code-point aliases.

Verifying the DSCP Classifier

Purpose Make sure that the DSCP classifier is configured as expected.

Action On Device R2, run the **show class-of-service classifiers name v4-ba-classifier** command.

```
user@R2> show class-of-service classifiers name v4-ba-classifier

Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class                  Loss priority
  000000             BE-data                           high
  000001             BE-data                           low
  000010             BE-data                           low
  000011             BE-data                           low
  000100             BE-data                           low
  000101             BE-data                           low
  000110             BE-data                           low
  000111             BE-data                           low
  001000             BE-data                           low
  001001             BE-data                           low
  001010             Voice                             low
  001011             BE-data                           low
  001100             Voice                             high
  001101             BE-data                           low
  001110             Voice                             high
  001111             BE-data                           low
  010000             BE-data                           low
  010001             BE-data                           low
  010010             BE-data                           low
  010011             BE-data                           low
  010100             BE-data                           low
  010101             BE-data                           low
  010110             BE-data                           low
  010111             BE-data                           low
  011000             BE-data                           low
  011001             BE-data                           low
  011010             BE-data                           low
  011011             BE-data                           low
  011100             BE-data                           low
  011101             BE-data                           low
  011110             BE-data                           low
```

```
  011111              BE-data                         low
  100000              BE-data                         low
  100001              BE-data                         low
  100010              BE-data                         low
  100011              BE-data                         low
  100100              BE-data                         low
  100101              BE-data                         low
  100110              BE-data                         low
  100111              BE-data                         low
  101000              BE-data                         low
  101001              BE-data                         low
  101010              BE-data                         low
  101011              BE-data                         low
  101100              BE-data                         low
  101101              BE-data                         low
  101110              Premium-data                    high
  101111              Premium-data                    low
  110000              NC                              low
  110001              BE-data                         low
  110010              BE-data                         low
  110011              BE-data                         low
  110100              BE-data                         low
  110101              BE-data                         low
  110110              BE-data                         low
  110111              BE-data                         low
  111000              NC                              low
  111001              BE-data                         low
  111010              BE-data                         low
  111011              BE-data                         low
  111100              BE-data                         low
  111101              BE-data                         low
  111110              BE-data                         low
  111111              BE-data                         low
```

Meaning Notice that the default classifier is incorporated into the customer classifier. If you were to remove the **import default** statement from the custom classifier, the custom classifier would look like this:

```
user@R2> show class-of-service classifier name v4-ba-classifier
Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point          Forwarding class                Loss priority
  000000              BE-data                         high
  000001              BE-data                         low
  101110              Premium-data                    high
  101111              Premium-data                    low
```

Verifying the Forwarding Classes and Output Queues

Purpose Make sure that the forwarding classes are configured as expected.

Action On Device R2, run the **show class-of-service forwarding-class** command.

```
user@R2> show class-of-service forwarding-class

Forwarding class                        ID      Queue  Restricted queue  Fabric
priority  Policing priority   SPU priority
  BE-data                                0       0          0             low
            normal             low
```

```
  Premium-data                    1       1       1            low
            normal          low
  Voice                           2       2       2            low
            normal          low
  NC                              3       3       3            low
            normal          low
```

Meaning The forwarding classes are configured as expected.

Verifying That the Classifier Is Applied to the Interfaces

Purpose Make sure that the classifier is applied to the correct interfaces.

Action On Device R2, run the **show class-of-service interface** command.

```
user@R2> show class-of-service interface ge-1/0/3

Physical interface: ge-1/0/3, Index: 144
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: ge-1/0/3.0, Index: 333
Object                  Name                    Type                    Index
Classifier              v4-ba-classifier        dscp                    10755

user@R2> show class-of-service interface ge-1/0/9

Physical interface: ge-1/0/9, Index: 150
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: ge-1/0/9.0, Index: 332
Object                  Name                    Type                    Index
Classifier              v4-ba-classifier        dscp                    10755
```

Meaning The interfaces are configured as expected.

Verifying Behavior Aggregate Classifiers

Purpose Verify that the behavior aggregate classifier is behaving as expected. Keep in mind that although the classifier operates on incoming packets, you view the resulting queue assignment on the outgoing interface.

Action

1. Clear the interface statistics on Device R2.

   ```
   user@R2> clear interface statistics ge-1/0/3
   ```

2. Using extended ping from Device R1 or a packet generator running on a host or server, send packets with the code point set to 101111.

   Both methods are shown here. The packet generator used is hping.

   - When you are using extended ping to set the DSCP code points in the IPv4 packet header, the ToS decimal value (in this case, 188) is required in the **tos** option of the **ping** command.

- When you are using hping to set the DSCP code points in the IPv4 packet header, the type-of-service (ToS) hex value (in this case, BC) is required in the **--tos** option of the **hping** command.

If your binary-to-hex or binary-to-decimal conversion skills are rusty, you can use an online calculator, such as http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html .

NOTE: When you convert a binary DSCP code point value, be sure to add two extra zeros at the end. So instead of 101111, use 10111100. These 0 values (the 7th and 8th bits) are reserved and ignored, but if you do not include them in the conversion, your hex and decimal values will be incorrect.

```
user@R1> ping 172.16.70.1 tos 188 rapid count 25

PING 172.16.70.1 (172.16.70.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!
--- 172.16.70.1 ping statistics ---
25 packets transmitted, 25 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.404/0.483/1.395/0.207 ms

root@host1> hping 172.16.70.1 --tos BC -c 25

HPING 172.16.70.1 (eth1 172.16.70.1): NO FLAGS are set, 40 headers + 0 data
bytes
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.3 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.6 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=0.6 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=0.3 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=7 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=20 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=21 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=22 win=0 rtt=0.4 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=23 win=0 rtt=0.5 ms
len=46 ip=172.16.70.1 ttl=61 DF id=0 sport=0 flags=RA seq=24 win=0 rtt=0.4 ms
```

3. On Device R2, verify that queue 2 is incrementing.

   Code point 101111 is associated with Premium-data, which uses queue 1.

   ```
   user@R2> show interfaces extensive ge-1/0/3 | find "queue counters"
   ```

```
Queue counters:   Queued packets   Transmitted packets   Dropped packets
  0                          0                     0                 0
  1                         50                    50                 0
  2                          0                     0                 0
  3                         42                    42                 0
  Queue number:         Mapped forwarding classes
    0                   BE-data
    1                   Premium-data
    2                   Voice
    3                   NC
...
```

Meaning The output shows that queue 1 has incremented by 50 packets after sending 50 packets through the router.

Related Documentation

- *Classification Overview*
- *Sample Behavior Aggregate Classification*
- *Understanding Packet Loss Priorities*

Example: Configuring DSCP Classification for VPLS

The following example configures DSCP classifier **dscp_vpls** on ATM interface **at-4/1/1** with **ether-vpls-over-atm-llc** encapsulation. The classifier **dscp_vpls** is applied to the interface and the interface is listed in the VPLS routing instance **vpls1** on the ingress PE router.

1. Configure the ATM interface **at-4/1/1.0** and the encapsulation as **ether-vpls-over-atm-llc**:

```
[edit]
interfaces {
  at-4/1/1 {
    mtu 9192;
    atm-options {
      vpi 10;
    }
    unit 0 {
      encapsulation ether-vpls-over-atm-llc;
      vci 10.128;
      family vpls;
    }
  }
}
```

2. Configure the DSCP classifier **dscp_vpls**:

```
[edit]
class-of-service {
  classifiers {
    dscp dscp_vpls {
      forwarding-class expedited-forwarding {
```

```
            loss-priority low code-points 000010;
        }
    }
  }
}
```

3. Apply the classifier **dscp_vpls** to the ATM interface **at-4/1/1.0**:

```
[edit]
interfaces {
  at-4/1/1 {
    unit 0 {
      classifiers {
        dscp dscp_vpls;
      }
    }
  }
}
```

4. Include the ATM interface virtual circuit **at-4/1/1.0** as part of the routing instance **vpls1** configuration:

```
[edit]
  routing-instances {
  vpls1 {
    instance-type vpls;
    interface at-4/1/1.0;
    route-distinguisher 10.255.245.51:1;
    vrf-target target:1234:1;
    protocols {
      vpls {
        site-range 10;
        no-tunnel-services;
        site vpls-1-site-1 {
          site-identifier 1;
        }
      }
    }
  }
}
```

Related Documentation

- Understanding DSCP Classification for VPLS on page 7

Example: DSCP IPv6 Rewrites and Forwarding Class Maps

You cannot configure a DSCP IPv6 rewrite rule and output forwarding class map on the same logical interface (unit). These must be used on different logical interfaces. Although a warning is issued, there is nothing in the CLI that prevents this configuration. An error message appears when you attempt to commit the configuration.

This example shows the warning and error message that results when the default DSCP IPv6 rewrite rule is configured on logical interface **ge-1/0/4.0** with output forwarding class map **vg1**.

```
[edit class-of-service]
```

```
interfaces {
  ge-1/0/4 {
    unit 0 {
      ##
      ## Warning: DSCP-IPv6 rewrite and forwarding class map not allowed on same unit
      ##
      output-forwarding-class-map vg1;
      rewrite-rules {
        dscp-ipv6 default;
      }
    }
  }
}
```

```
user@router# commit
[edit class-of-service interfaces ge-1/0/4 unit 0 output-forwarding-class-map]
  'output-forwarding-class-map vg1'
    DSCP-IPv6 rewrite and forwarding class map not allowed on same unit
error: commit failed: (statements constraint check failed)
```

Related Documentation

- *Applying Forwarding Classes to Interfaces*

Tunneling and BA Classifiers

BA classifiers can be used with GRE and IP-IP tunnels on the following routers and switches:

- EX Series switches
- M7i and M10i routers
- M Series routers with E-FPC or EP-FPC
- M120 routers
- M320 routers
- T Series routers

When a GRE or IP-IP tunnel is configured on an incoming (core-facing) interface, the queue number and PLP information are carried through the tunnel. At the egress (customer-facing) interface, the packet is queued and the CoS bits rewritten based on the information carried through the tunnel.

If no BA classifier is configured in the incoming interface, the default classifier is applied. If no rewrite rule is configured, the default rewrite rule is applied.

Configuring Classifiers and Rewrite Rules at the Global and Physical Interface Levels

On ACX Series Universal Access Routers and EX Series switches, CoS supports classification and rewrite at the global and physical interface levels.

To configure the global EXP classifier, include the following statements at the **[edit class-of-service] system-defaults** hierarchy level.

```
[edit class-of-service]
{
  system-defaults
    {
    classifiers exp classifier-name
  }
}
```

CoS supports one global system default classifier of the EXP type, as shown in the following example:

```
[edit class-of-service]
{
  system-defaults {
    classifiers {
      exp exp-classf-core;
    }
  }
}
```

To configure classifiers and rewrite rules at the physical interface level, include the following statements at the **[edit class-of-service] interfaces** hierarchy level.

```
[edit class-of-service]
interfaces {
  interface-name
    classifiers dscp classifier-name
    classifiers inet-precedence classifier-name
    classifiers ieee-802.1 [vlan-tag (outer | inner)] classifier-name
    rewrite-rules dscp rewrite-name
    rewrite-rules inet-prec rewrite-name
    rewrite-rules ieee-802.1 rewrite-name
}
```

The following example shows classifiers and rewrite rules configured on physical interfaces:

```
ge-0/1/0 {
  unit 0 {
    rewrite-rules {
      exp custom-exp;
    }
  }
  classifiers {
    dscp d1;
    ieee-802.1 ci;
  }
  rewrite-rules {
    dscp default;
  }
}
  ge-0/1/2 {
    classifiers {
      ieee-802.1 ci;
    }
    rewrite-rules {
      ieee-802.1 ri;
```

```
    }
  }
  ge-0/1/3 {
    unit 0 {
      rewrite-rules {
        exp custom-exp2;
      }
    }
  }
  ge-0/1/7 {
    classifiers {
      dscp d1;
    }
  }
  ge-0/1/8 {
    classifiers {
      dscp d1;
    }
  }
```

Related Documentation

- Classifiers and Rewrite Rules at the Global and Physical Interface Levels Overview on page 8

CHAPTER 5

Configuration Task for DSCP IPv6 Classifiers

- Applying DSCP IPv6 Classifiers on page 57

Applying DSCP IPv6 Classifiers

For M320 and T Series routers and EX Series switches, you can apply separate classifiers for IPv4 and IPv6 packets per logical interface by including the **classifiers** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level and specifying the **dscp** and **dscp-ipv6** classifier types:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
classifiers dscp (classifier-name | default) family (mpls | inet);
classifiers dscp-ipv6 (classifier-name | default) family (mpls | inet));
```

For M Series router enhanced FPCs, you cannot apply separate classifiers for IPv4 and IPv6 packets on a single logical interface. Instead, classifier assignment works as follows:

- If you assign a DSCP classifier only, IPv4 and IPv6 packets are classified using the DSCP classifier.
- If you assign an IP precedence classifier only, IPv4 and IPv6 packets are classified using the IP precedence classifier. In this case, the lower three bits of the DSCP field are ignored because IP precedence mapping requires the upper three bits only.
- If you assign either the DSCP or the IP precedence classifier in conjunction with the DSCP IPv6 classifier, the commit fails.
- If you assign a DSCP IPv6 classifier only, IPv4 and IPv6 packets are classified using the DSCP IPv6 classifier, but the commit displays a warning message.

For more information, see "Applying Classifiers to Logical Interfaces" on page 24. For a complex configuration example, see the *Junos OS, Release 14.1*.

CHAPTER 6

Configuration Tasks for MPLS EXP Classifiers

- Applying MPLS EXP Classifiers to Routing Instances on page 59
- Applying MPLS EXP Classifiers for Explicit-Null Labels on page 63

Applying MPLS EXP Classifiers to Routing Instances

When you enable VRF table labels and you do not explicitly apply a classifier configuration to the routing instance, the default MPLS EXP classifier is applied to the routing instance. For detailed information about VRF table labels, see the *Junos OS VPNs Library for Routing Devices*.

The default MPLS EXP classification table contents are shown in Table 12 on page 59.

Table 12: Default MPLS EXP Classification 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

For PICs that are installed on enhanced FPCs, you can override the default MPLS EXP classifier and apply a custom classifier to the routing instance. To do this, perform the following configuration tasks:

1. Filter traffic based on the IP header by including the **vrf-table-label** statement at the **[edit routing-instances *routing-instance-name*]** hierarchy level:

   ```
   [edit routing-instances routing-instance-name]
   vrf-table-label;
   ```

2. Configure a custom MPLS EXP classifier by including the following statements at the **[edit class-of-service]** hierarchy level:

   ```
   [edit class-of-service]
   classifiers {
     exp classifier-name {
       import (classifier-name | default);
       forwarding-class class-name {
         loss-priority level code-points [ aliases ] [ bit-patterns ];
       }
     }
   }
   forwarding-classes {
     queue queue-number class-name priority (high | low);
   }
   ```

3. Configure the routing instance to use the custom MPLS EXP classifier by including the **exp** statement at the **[edit class-of-service routing-instances *routing-instance-name* classifiers]** hierarchy level:

   ```
   [edit class-of-service routing-instances routing-instance-name classifiers]
   exp classifier-name;
   ```

To display the MPLS EXP classifiers associated with all routing instances, issue the **show class-of-service routing-instances** command.

NOTE: The following caveats apply to custom MPLS EXP classifiers for routing instances:

- An enhanced FPC is required.
- Logical systems are not supported.

For more details, see the following sections:

- Configuring Global Classifiers and Wildcard Routing Instances on page 60
- Examples: Applying MPLS EXP Classifiers to Routing Instances on page 61

Configuring Global Classifiers and Wildcard Routing Instances

To configure a global routing instance classifier, include the **all** statement at the **[edit class-of-service routing-instances]** hierarchy level:

```
[edit class-of-service routing-instances]
all {
  classifiers {
    exp classifier-name;
  }
```

```
}
```

For routing instances associated with specific classifiers, the global configuration is ignored.

To use a wildcard in the routing instance classifier configuration, include an asterisk (*) in the name of the routing instance:

```
[edit class-of-service routing-instances]
routing-instance-name* {
  classifiers {
    exp classifier-name;
  }
}
```

The wildcard configuration follows the longest match. If there is a specific configuration, it is given precedence over the wildcard configuration.

NOTE: Wildcards and the all keyword are supported at the [edit class-of-service routing-instances] hierarchy level but not at the [edit routing-instances] hierarchy level.

If you configure a routing instance at the [edit routing-instances] hierarchy level with, for example, the name vpn*, the Junos OS treats vpn* as a valid and distinct routing instance name. If you then try to apply a classifier to the vpn* routing instance at the [edit class-of-service routing-instances] hierarchy level, the Junos OS treats the vpn* routing instance name as a wildcard, and all the routing instances that start with vpn and do not have a specific classifier applied receive the classifier associated with vpn*. This same behavior applies with the all keyword.

Examples: Applying MPLS EXP Classifiers to Routing Instances

Configure a global classifier for all routing instances and override the global classifier for a specific routing instance. In this example, there are three routing instances: **vpn1**, **vpn2**, and **vpn3**, each with VRF table label enabled. The classifier **exp-classifier-global** is applied to **vpn1** and **vpn2** (that is, all but **vpn3**, which is listed separately). The classifier **exp-classifier-3** is applied to **vpn3**.

Configuring a Global Classifier

```
[edit routing-instances]
vpn1 {
  vrf-table-label;
}
vpn2 {
  vrf-table-label;
}
vpn3 {
  vrf-table-label;
}

[edit class-of-service routing-instances]
all {
```

```
    classifiers {
      exp exp-classifier-global;
    }
  }
  vpn3 {
    classifiers {
      exp exp-classifier-3;
    }
  }
```

Configure a wildcard routing instance and override the wildcard with a specific routing instance. In this example, there are three routing instances: **vpn-red**, **vpn-yellow**, and **vpn-green**, each with VRF table label enabled. The classifier **exp-class-wildcard** is applied to **vpn-yellow** and **vpn-green**. The classifier **exp-class-red** is applied to **vpn-red**.

Configuring a Wildcard Routing Instance

```
[edit routing-instances]
vpn-red {
  vrf-table-label;
}
vpn-yellow {
  vrf-table-label;
}
vpn-green {
  vrf-table-label;
}

[edit class-of-service routing-instances]
vpn* {
  classifiers {
    exp exp-class-wildcard;
  }
}
vpn-red {
  classifiers {
    exp exp-class-red;
  }
}
```

Display the MPLS EXP classifiers associated with two routing instances:

Monitoring a Configuration

```
[edit class-of-service routing-instances]
vpn1 {
  classifiers {
    exp default;
  }
}
vpn2 {
  classifiers {
    exp class2;
  }
}
```

```
user@host> show class-of-service routing-instances
  Routing Instance : vpn1
    Object           Name                  Type                  Index
    Classifier       exp-default           exp                       8
```

```
Routing Instance : vpn2
  Object           Name                 Type                 Index
  Classifier       class2               exp                  57507
```

Applying MPLS EXP Classifiers for Explicit-Null Labels

When you configure MPLS explicit-null labels, label 0 is advertised to the egress router of an LSP. When label 0 is advertised, the egress router (instead of the penultimate router) removes the label. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label. For more information about explicit-null labels and ultimate-hop popping, see the *Junos OS MPLS Applications Library for Routing Devices*.

On M320 and T Series routers, when you configure MPLS explicit-null labels with an MPLS EXP classifier, the MPLS EXP classifier can be different from an IPv4 or IPv6 classifier configured on the same logical interface. In other words, you can apply separate classifiers for MPLS EXP, IPv4, and IPv6 packets per logical interface. To combine an EXP classifier with a distinct IPv6 classifier, the PIC must be mounted on an Enhanced FPC.

NOTE: For J Series routers and other M Series routers, MPLS explicit-null labels with MPLS EXP classification are supported if you set the same classifier for EXP and IPv4 traffic, or EXP and IPv6 traffic.

For more information about how IPv4 and IPv6 packet classification is handled, see "Applying DSCP IPv6 Classifiers" on page 57.

To configure an MPLS EXP classifiers for explicit-null labels, include the **exp** statement at the **[edit class-of-service classifiers]** and **[edit class-of-service interfaces *interface-name* unit *logical-unit-number* classifiers]** hierarchy levels:

```
[edit class-of-service classifiers]
exp classifier-name {
   import (classifier-name | default);
   forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
   }
}
[edit class-of-service interfaces interface-name unit logical-unit-number classifiers]
exp (classifier-name | default);
```

CHAPTER 7

Configuration Task for IEEE 802.1ad Classifiers

- Configuring and Applying IEEE 802.1ad Classifiers on page 65

Configuring and Applying IEEE 802.1ad Classifiers

For Juniper Network MX Series 3D Universal Edge Router interfaces or IQ2 PICs with IEEE 802.1ad frame formats or EX Series switches, you can set the forwarding class and loss priority for traffic on the basis of the three IEEE 802.1p bits and the DEI bit. You can apply the default map or customize one or more of the default values.

You then apply the classifier to the interface on which you configure IEEE 802.1ad frame formats.

Defining Custom IEEE 802.1ad Maps

You can customize the default IEEE 802.1ad map by defining values for IEEE 802.1ad code points.

```
class-of-service {
  classifiers {
    ieee-802.1ad dot1p_dei_class {
      forwarding-class best-effort {
        loss-priority low code-points [ 0000 1101 ];
      }
    }
  }
}
```

Applying Custom IEEE 802.1ad Maps

You then apply the classifier map to the logical interface:

```
interfaces {
  ge-2/0/0 {
    unit 0 {
      classifiers {
        ieee-802.1ad dot1p_dei_class;
      }
    }
  }
```

```
}
```

Verifying Custom IEEE 802.1ad Map Configuration

To verify your configuration, you can issue the following operational mode commands:

- **show class-of-service forwarding-table loss-priority-map**
- **show class-of-service forwarding-table loss-priority-map mapping**
- **show chassis forwarding**
- **show pfe fwdd**

CHAPTER 8

Configuration Statements

- BA Classifier Configuration Hierarchy on page 68
- classifiers (Logical Interface) on page 69
- classifiers (Routing Instance) on page 70
- classifiers (Definition) on page 71
- classifiers (Physical Interface) on page 72
- code-point-aliases on page 73
- code-points on page 73
- dscp (AS PIC Classifiers) on page 74
- dscp (Classifier on Physical Interface) on page 74
- dscp (Class of Service Classifier) on page 75
- dscp (Rewrite Rules) on page 76
- dscp-ipv6 (Class-of-Service) on page 77
- exp on page 78
- forwarding-class (BA Classifiers) on page 79
- ieee-802.1 (Classifier on Physical Interface) on page 79
- ieee-802.1 (Rewrite Rules on Logical Interface) on page 80
- ieee-802.1ad on page 81
- import (Classifiers) on page 82
- inet-precedence on page 82
- inet-precedence (Classifier on Physical Interface) on page 83
- interfaces on page 84
- loss-priority (BA Classifiers) on page 86
- routing-instances (Class of Service) on page 87
- system-defaults on page 88
- unit on page 89

BA Classifier Configuration Hierarchy

To configure BA classifiers, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
classifiers {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence) classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
interfaces {
  interface-name {
    unit logical-unit-number {
      classifiers {
        (dscp | dscp-ipv6 | exp | ieee-802.1 | ieee-802.1ad | inet-precedence)
          (classifier-name | default);
      }
    }
  }
}
routing-instances routing-instance-name {
  classifiers {
    exp (classifier-name | default);
  }
}
```

classifiers (Logical Interface)

Syntax

```
classifiers {
  type (classifier-name | default) family (mpls | inet);
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a CoS aggregate behavior classifier to a logical interface. You can apply a default classifier or one that is previously defined.

Options ***classifier-name***—Name of the aggregate behavior classifier.

type—Traffic type.

Values: dscp, dscp-ipv6, exp, ieee-802.1, inet-precedence

NOTE: You can only specify a family for the dscp and dscp-ipv6 types.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Default DSCP and DSCP IPv6 Classifier on page 14
- Applying Classifiers to Logical Interfaces on page 24

classifiers (Routing Instance)

Syntax

```
classifiers {
  exp (classifier-name | default);
  dscp (classifier-name | default);
  dscp-ipv6 (classifier-name | default);
}
```

Hierarchy Level [edit class-of-service routing-instances *routing-instance-name*]

Release Information Statement introduced before Junos OS Release 7.4.
dscp and **dscp-ipv6** support introduced in Junos OS Release 9.6.

Description For routing instances with VRF table labels enabled, apply a custom Multiprotocol Label Switching (MPLS) EXP classifier or DSCP classifier to the routing instance. You can apply the default classifier or one that is previously defined.

Options ***classifier-name***—Name of the behavior aggregate MPLS EXP or DSCP classifier.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Applying MPLS EXP Classifiers to Routing Instances on page 59
- Applying Classifiers to Logical Interfaces on page 24

classifiers (Definition)

Syntax

```
classifiers {
  type classifier-name {
    import (classifier-name | default);
    forwarding-class class-name {
      loss-priority level code-points [ aliases ] [ bit-patterns ];
    }
  }
}
```

Hierarchy Level

[edit class-of-service],
[edit class-of-service routing-instances *routing-instance-name*]

Release Information

Statement introduced before Junos OS Release 7.4.
ieee-802.1ad option introduced in Junos OS Release 9.2.

Description

Define a CoS behavior aggregate (BA) classifier for classifying packets. You can associate the classifier with a forwarding class or code-point mapping, and import a default classifier or one that is previously defined.

NOTE: The **[edit class-of-service routing-instances *routing-instance-name*]** hierarchy level and the **dscp-ipv6** and **ieee-802.1ad** classifier types are not supported on ACX Series routers.

Options

classifier-name—Name of the aggregate behavior classifier.

type—Traffic type: **dscp**, **dscp-ipv6**, **exp**, **ieee-802.1**, **ieee-802.1ad**, **inet-precedence**.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 5

classifiers (Physical Interface)

Syntax

```
classifiers {
  type (classifier-name | default) ;
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access routers.

Description Apply a CoS aggregate behavior classifier to a physical interface. You can apply a default classifier or one that is previously defined.

Options

classifier-name—Name of the aggregate behavior classifier.

type—Traffic type.

Values: **dscp**, **ieee-802.1**, and **inet-precedence**

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- dscp on page 74
- inet-precedence on page 83
- ieee-802.1 on page 79

code-point-aliases

Syntax

```
code-point-aliases {
  type {
    alias-name bits;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define an alias for a CoS marker.

Options

alias-name—Name of the code-point alias.

bits—6-bit value of the code-point bits, in decimal form.

type—CoS marker type.

Values: **dscp**, **dscp-ipv6**, **exp**, **ieee-802.1**, **inet-precedence**

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining Code Point Aliases for Bit Patterns on page 22

code-points

Syntax code-points ([*aliases*] | [*bit-patterns*]);

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name* loss-priority *level*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for SRX Series devices.

Description Specify one or more DSCP code-point aliases or bit sets for association with a forwarding class.

Options

aliases—Name of the DSCP alias.

bit-patterns—Value of the code-point bits, in six-bit binary form.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 5
- Example: Configuring a Custom DSCP Behavior Aggregate Classifier on page 39

dscp (AS PIC Classifiers)

Syntax dscp (*alias* | *bits*);

Hierarchy Level [edit services cos application-profile *profile-name* (ftp | sip) (data | video | voice)],
[edit services cos rule *rule-name* term *term-name* then],
[edit services cos rule *rule-name* term *term-name* then (reflexive | reverse)]

Release Information Statement introduced in Junos OS Release 8.1.

Description Define the Differentiated Services code point (DSCP) mapping that is applied to the packets.

Options ***alias***—Name assigned to a set of CoS markers.

bits—Mapping value in the packet header.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Configuring Actions in a CoS Rule*

dscp (Classifier on Physical Interface)

Syntax dscp (*classifier-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name*classifiers]

Release Information Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access routers.

Description For ACX Series Universal Access routers, map the DSCP field of the incoming packet to the forwarding class and packet loss priority based on the specified DSCP classifier.

Options ***classifier-name***—Name of the previously defined DSCP behavior aggregate classifier.

default—Default DSCP behavior aggregate classifier.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dscp (Class of Service Classifier)

Syntax

```
dscp classifier-name {
  import (classifier-name | default);
  forwarding-class class-name {
    loss-priority level code-points [ aliases ] [ bit-patterns ]
    }
  }
```

Hierarchy Level [edit class-of-service *classifiers*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the diffserv code point (DSCP) mapping that is applied to the packets.

Options ***classifier-name***—Name of the classifier.

The remaining statements are explained separately.

Required Privilege Level interfaces—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- BA Classifier Overview on page 3
- BA Classifier Configuration Hierarchy on page 68
- Overview of BA Classifier Types on page 5
- DSCP Classifier Configuration Examples on page 28

dscp (Rewrite Rules)

Syntax dscp (*rewrite-name* | default) protocol mpls;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.

Description For IPv4 traffic, apply a Differentiated Services (DiffServ) code point (DSCP) rewrite rule.

Logical interfaces do not support multiple **dscp** rewrite rules for the same protocol.

DSCP and DSCP IPv6 rewrite rules are supported on M Series and T Series routers when non-queuing PICs are installed, but are disabled when queuing PICs are installed with the following exceptions:

- On M320 routers, DSCP rewrite is supported on IQ, IQ2, IQE, and IQ2E PICs when used with the Enhanced III FPC.
- On M120 routers, DSCP rewrite is supported on IQ, IQ2, IQE, and IQ2E PICs.

DSCP and DCSP IPv6 rewrite rules are supported on MIC and MPC interfaces on MX Series routers.

DSCP rewrite rules are not supported on T Series routers when IQ, IQ2, IQE, IQ2E, SONET/SDH OC48/STM16 IQE, or PD-5-10XGE-SFPP PICs are installed.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules dscp]** hierarchy level.

default—The default mapping.

protocol mpls—(Optional for ingress MPLS tunnel nodes) For interfaces on MX Series routers or hosted on Enhanced III FPCs in M120 or M320 routers only, rewrite the MPLS EXP bits in the MPLS header independently of the IPv4 DSCP value for IPv4 packets entering an MPLS tunnel.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Rewrite Rules*
- *Applying Rewrite Rules to Output Logical Interfaces*
- *protocol (Rewrite Rules)*
- *Rewriting MPLS and IPv4 Packet Headers*
- *rewrite-rules (Definition)*

dscp-ipv6 (Class-of-Service)

Syntax dscp-ipv6 (*rewrite-name* | <default>) protocol mpls;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.
Support for **protocol mpls** option introduced in Junos OS Release 10.4R2.

Description For IPv6 traffic, apply a DSCP rewrite rule.

Logical interfaces do not support multiple **dscp-ipv6** rewrite rules for the same protocol.

DSCP and DSCP IPv6 rewrite rules are supported on M Series and T Series routers when non-queuing PICs are installed, but are disabled when queuing PICs are installed with the following exceptions:

- On M320 routers, DSCP rewrite is supported on IQ, IQ2, IQE, and IQ2E PICs when used with the Enhanced III FPC.
- On M120 routers, DSCP rewrite is supported on IQ, IQ2, IQE, and IQ2E PICs.

DSCP and DCSP IPv6 rewrite rules are supported on MIC and MPC interfaces on MX Series routers.

DSCP rewrite rules are not supported on T Series routers when IQ, IQ2, IQE, IQ2E, SONET/SDH OC48/STM16 IQE, or PD-5-10XGE-SFPP PICs are installed.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules dscp-ipv6]** hierarchy level.

default—Default mapping.

protocol mpls—(Optional for ingress MPLS tunnel nodes) For interfaces on MX Series routers or hosted on Enhanced III FPCs in M120 or M320 routers only, rewrite the MPLS EXP bits in the MPLS header independently of the IPv6 DSCP value for IPv6 packets entering an MPLS tunnel.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Rewrite Rules*
- *protocol*
- *Setting IPv6 DSCP and MPLS EXP Values Independently*
- *Configuring DSCP Values for IPv6 Packets Entering the MPLS Tunnel*
- *Applying Rewrite Rules to Output Logical Interfaces*
- *rewrite-rules (Definition)*

exp

Syntax exp (*rewrite-name* | default) protocol *protocol-types*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced before Junos OS Release 12.2. for ACX series

Description Apply an MPLS experimental (EXP) rewrite rule.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules exp]** hierarchy level.

default—The default mapping.

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. 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 *interface-name* unit *logical-unit-number* rewrite-rules exp *rewrite-rule-name* protocol]** hierarchy level. The IP precedence rewrite rules explanation does not apply to ACX Series Universal Access routers.

On interfaces configured on Modular Port Concentrators (MPCs) and Modular Interface Cards (MICs) on MX Series 3D Universal Edge Routers and EX Series switches, we highly recommend that you configure the **default** option when you configure a behavior aggregate (BA) classifier that does not include a specific rewrite rule for MPLS packets. Doing so ensures that MPLS exp value is rewritten according to the BA classifier rules configured for forwarding or packet loss priority. This does not apply to ACX Series Universal Access routers.

protocol-types—Specify one or more protocol matching criteria:

- **mpls-any**—Apply to MPLS packets, write MPLS header only.
- **mpls-inet-both**—Apply to IPv4 MPLS packets, write MPLS and IPv4 header.
- **mpls-inet-both-non-vpn**—Apply to IPv4 MPLS packets, write MPLS and IPv4 header for only non VPN traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Rewrite Rules*
- *Rewriting the EXP Bits of All Three Labels of an Outgoing Packet*
- *Applying Rewrite Rules to Output Logical Interfaces*
- *protocol (Rewrite Rules)*

- *rewrite-rules (Definition)*

forwarding-class (BA Classifiers)

Syntax

```
forwarding-class class-name {
  loss-priority level code-points [ aliases ] [ bit-patterns ];
}
```

Hierarchy Level [edit class-of-service classifiers *type classifier-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define forwarding class name and option values.

Options ***class-name***—Name of the forwarding class.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Defining Classifiers on page 21

ieee-802.1 (Classifier on Physical Interface)

Syntax ieee-802.1 (*classifier-name* | default) vlan-tag (inner | outer);

Hierarchy Level [edit class-of-service interfaces *interface-name*classifiers]

Release Information Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access routers.

Description For ACX Series Universal Access routers and EX Series switches, map the ieee-802.1p field of the incoming packet to the forwarding class and packet loss priority based on the specified 802.1p classifier. In the case of double tagged packets, you can configure whether to use the 802.1p of the outer or inner VLAN tag.

Options **vlan-tag inner**—In the case of double tagged packets, classify based on the 802.1p of the inner VLAN tag.

vlan-tag outer—Classify based on the 802.1p of the outermost VLAN tag.

classifier-name—Name of the previously defined ieee-802.1p behavior aggregate classifier.

default—Default ieee-802.1p behavior aggregate classifier.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

ieee-802.1 (Rewrite Rules on Logical Interface)

Syntax ieee-802.1 (*rewrite-name* | default) vlan-tag (outer | outer-and-inner);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.
vlan-tag statement introduced in Junos OS Release 8.1.

Description Apply an IEEE-802.1 rewrite rule. For IQ PICs, you can only configure one IEEE 802.1 rewrite rule on a physical port. All logical ports (units) on that physical port should apply the same IEEE 802.1 rewrite rule.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules ieee-802.1]** hierarchy level.

default—The default mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Rewrite Rules*
- dscp (Rewrite Rules) on page 76
- dscp-ipv6 (Class-of-Service) on page 77
- exp on page 78
- *exp-push-push-push*
- *exp-swap-push-push*
- ieee-802.1ad on page 81
- inet-precedence on page 82
- *rewrite-rules (Definition)*

ieee-802.1ad

Syntax ieee-802.1ad (*rewrite-name* | default) vlan-tag (outer | outer-and-inner);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced in Junos OS Release 9.2.

Description Apply a IEEE-802.1ad rewrite rule.

Options

rewrite-name—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules ieee-802.1ad]** hierarchy level.

default—The default rewrite bit mapping.

vlan-tag—The rewrite rule is applied to the **outer** or **outer-and-inner** VLAN tag.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Rewrite Rules*
- dscp (Rewrite Rules) on page 76
- dscp-ipv6 (Class-of-Service) on page 77
- exp on page 78
- *exp-push-push-push*
- *exp-swap-push-push*
- ieee-802.1 (Rewrite Rules on Logical Interface) on page 80
- inet-precedence on page 82
- *rewrite-rules (Definition)*

import (Classifiers)

Syntax import (*classifier-name* | default);

Hierarchy Level [edit class-of-service classifiers *type classifier-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify a default or previously defined classifier.

Options ***classifier-name***—Name of the classifier mapping configured at the **[edit class-of-service classifiers]** hierarchy level.

default—The default classifier mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Overview of BA Classifier Types on page 5

inet-precedence

Syntax inet-precedence (*rewrite-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules]

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a IPv4 precedence rewrite rule.

Options ***rewrite-name***—Name of a **rewrite-rules** mapping configured at the **[edit class-of-service rewrite-rules inet-precedence]** hierarchy level.

default—The default mapping. 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.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Configuring Rewrite Rules*
- *Applying Rewrite Rules to Output Logical Interfaces*
- *protocol (Rewrite Rules)*
- *rewrite-rules (Definition)*

inet-precedence (Classifier on Physical Interface)

Syntax inet-precedence (*classifier-name* | default);

Hierarchy Level [edit class-of-service interfaces *interface-name*classifiers]

Release Information Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access routers.

Description On ACX Series Universal Access routers and EX Series switches, map the inet-precedence field of the incoming packet to the forwarding class and packet loss priority, based on the specified inet-precedence classifier. When no classifier is configured on the physical interface, the default ipprec-compatibility inet-precedence classifier is applied on the physical interface.

Options ***classifier-name***—Name of the previously defined inet-precedence behavior aggregate classifier.

default—Default inet-precedence behavior aggregate classifier.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

interfaces

Syntax

```
interfaces {
  interface-name {
    classifiers{
      dscp(classifier-name | default) {
    }
      ieee-802.1 (classifier-name | default)  vlan-tag (inner | outer | classifier-name);
      inet-precedence (rewrite-name | default);
    }
    input-scheduler-map map-name;
    input-shaping-rate rate;
    irb {
      unit logical-unit-number {
        classifiers {
          type (classifier-name | default);
        }
        rewrite-rules {
          dscp (rewrite-name | default);
          dscp-ipv6 (rewrite-name | default);
          exp (rewrite-name | default)protocol protocol-types;
          ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
          inet-precedence (rewrite-name | default);
        }
      }
    }
    member-link-scheduler (replicate | scale);
    rewrite-rules {
      dscp (rewrite-name | default);
      ieee-802.1 (rewrite-name | default) vlan-tag (outer);
      inet-precedence (rewrite-name | default);
      }
    }
    scheduler-map map-name;
    scheduler-map-chassis map-name;
    shaping-rate rate;
    unit logical-unit-number {
      classifiers {
        type (classifier-name | default) family (mpls | inet);
      }
      forwarding-class class-name;
      fragmentation-map map-name;
      input-shaping-rate (percent percentage | rate);
      input-traffic-control-profile profile-name shared-instance instance-name;
      output-traffic-control-profile profile-name shared-instance instance-name;
      per-session-scheduler;
      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);
```

```
        }
        scheduler-map map-name;
        shaping-rate rate;
        translation-table (to-dscp-from-dscp | to-dscp-ipv6-from-dscp-ipv6 | to-exp-from-exp
          | to-inet-precedence-from-inet-precedence) table-name;
      }
    }
    interface-set interface-set-name {
      excess-bandwidth-share;
      internal-node;
      output-traffic-control-profile profile-name;
      output-traffic-control-profile-remaining profile-name;
    }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.
Interface-set level added in Junos OS Release 8.5.

Description Configure interface-specific CoS properties for incoming packets.

Options The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 5
- *Configuring Rewrite Rules*

loss-priority (BA Classifiers)

Syntax loss-priority *level*;

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify packet loss priority value for a specific set of code-point aliases and bit patterns.

Options ***level*** can be one of the following:

- **high**—Packet has high loss priority.
- **medium-high**—Packet has medium-high loss priority.
- **medium-low**—Packet has medium-low loss priority.
- **low**—Packet has low loss priority.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 5
- *Configuring Tricolor Marking*

routing-instances (Class of Service)

Syntax

```
routing-instances routing-instance-name {
  classifiers {
    dscp (classifier-name | default);
    dscp-ipv6 (classifier-name | default);
    exp (classifier-name | default);
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description For routing instances with VRF table labels enabled, apply a custom MPLS EXP classifier or DSCP classifier to the routing instance. You can apply the default MPLS EXP classifier or one that is previously defined.

Default If you do not include this statement, the default MPLS EXP classifier is applied to the routing instance. When no DSCP classifier is configured, the default MPLS EXP classifier is applied.

Options ***routing-instance-name***—Name of a routing instance.

classifier-name—Name of the behavior aggregate MPLS EXP classifier or DSCP classifier.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Forwarding Classes*
- *Applying Custom MPLS EXP Classifiers to Routing Instances in Layer 3 VPNs*

system-defaults

Syntax

```
system-defaults {
  classifiers{
    type classifier-name;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS release 12.2

Description Define a CoS classifier to support global classifiers.

Options ***classifier-name***—Name of the behavior aggregate (BA) classifier.

type—Traffic type: dscp, dscp-ipv6, or exp.

NOTE: The **dscp** and **dscp-ipv6** classifier types are not supported on ACX Series routers.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

unit

Syntax

```
unit logical-unit-number {
  classifiers {
    type (classifier-name | default) family (mpls | all);
  }
  forwarding-class class-name;
  fragmentation-map map-name;
  input-traffic-control-profile profile-name shared-instance instance-name;
  output-traffic-control-profile profile-name shared-instance instance-name;
  per-session-scheduler;
  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);
  }
  scheduler-map map-name;
  shaping-rate rate;
}
```

Hierarchy Level [edit class-of-service interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure a logical interface on the physical device. You must configure a logical interface to be able to use the physical device.

Options ***logical-unit-number***—Number of the logical unit.

Range: 0 through 16,384

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Overview of BA Classifier Types on page 5
- *Configuring Rewrite Rules*

PART 3

Administration

- Operational Commands on page 93

CHAPTER 9

Operational Commands

- show class-of-service code-point-aliases
- show class-of-service translation-table

show class-of-service code-point-aliases

Syntax show class-of-service code-point-aliases
<dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display the mapping of class-of-service (CoS) code point aliases to corresponding bit patterns.

Options **none**—Display code point aliases of all code point types.

dscp—(Optional) Display Differentiated Services code point (DSCP) aliases.

dscp-ipv6—(Optional) Display IPv6 DSCP aliases.

exp—(Optional) Display MPLS EXP code point aliases.

ieee-802.1—(Optional) Display IEEE-802.1 code point aliases.

inet-precedence—(Optional) Display IPv4 precedence code point aliases.

Required Privilege Level view

List of Sample Output show class-of-service code-point-aliases exp on page 95

Output Fields Table 13 on page 94 describes the output fields for the **show class-of-service code-point-aliases** command. Output fields are listed in the approximate order in which they appear.

Table 13: show class-of-service code-point-aliases Output Fields

Field Name	Field Description
Code point type	Type of the code points displayed: **dscp**, **dscp-ipv6** (not on EX Series switch), **exp** (not on EX Series switch or the QFX Series), **ieee-802.1**, or **inet-precedence** (not on the QFX Series).
Alias	Alias for a bit pattern.
Bit pattern	Bit pattern for which the alias is displayed.

Sample Output

show class-of-service code-point-aliases exp

```
user@host> show class-of-service code-point-aliases exp
Code point type: exp
  Alias               Bit pattern
  af11                100
  af12                101
  be                  000
  be1                 001
  cs6                 110
  cs7                 111
  ef                  010
  ef1                 011
  nc1                 110
  nc2                 111
```

show class-of-service translation-table

Syntax

```
show class-of-service translation–table
<name translation-table-name> |
<type (to-dscp–from–dscp | to-dscp-ipv6–from–dscp–ipv6 | to-exp–from–exp |
  to-inet-precedence-from-inet-precedence)>
```

Release Information Command introduced in Junos OS Release 9.3 for IQE PICs.

Description Display the mapping of class-of-service (CoS) translation table code points to corresponding bit patterns.

Options

none—Display translation table code points for all translation tables.

name—(Optional) Display information for the named translation table.

type—(Optional) Display information for a certain translation table type:

- **to-dscp-from-dscp**—Display DSCP translation table information.
- **to-dscp-ipv6-from-dscp-ipv6**—Display DSCP IPv6 translation table information.
- **to-exp-from-exp**—Display MPLS EXP translation table information.
- **to-inet-precedence-from-intet-precedence**—Display Internet precedence translation table information.

Required Privilege Level view

List of Sample Output

show class-of-service translation-table on page 97
show class-of-service translation-table name exp-trans-table on page 98
show class-of-service translation-table type to-dscp-ipv6-from-dscp-ipv6 on page 98

Output Fields Table 14 on page 96 describes the output fields for the **show class-of-service translation-table** command. Output fields are listed in the approximate order in which they appear.

Table 14: show class-of-service translation-table Output Fields

Field Name	Field Description
Translation Table	Name of the translation table.
Translation table type	Type of the translation table.
Index	Internal index number of the translation table.
From Code Point	Value of code point received.
To Code Point	Value of translated code point.

Sample Output

show class-of-service translation-table

```
user@host> show class-of-service translation-table
Translation Table: inet-trans-table, Translation table type: inet-to-inet, Index:
 61075
  From Code point    To Code Point
  000                          101
  001                          111
  010                          101
  011                          111
  100                          101
  101                          101
  110                          001
  111                          000

Translation Table: dscp-trans-table, Translation table type: dscp-to-dscp, Index:
 6761
  From Code point    To Code Point
  000000                    000111
  000001                    000111
  000010                    000111
  000011                    000111
  000100                    000111
  000101                    000111
  000110                    000111
  000111                    111000
  001000                    000111
  001001                    000111
  001010                    000111
  001011                    000111
  001100                    000111
  001101                    000111
  001110                    000111
  001111                    000111
  010000                    000111
  010001                    000111
  010010                    000111
  010011                    000111
  010100                    000111
  010101                    000111
  010110                    000111
  010111                    000111
  011000                    000111
  011001                    000111
  011010                    000111
  011011                    000111
  011100                    000111
  011101                    000111
  011110                    000111
  011111                    000111
  100000                    000111
  100001                    000111
  100010                    000111
  100011                    000111
  100100                    000111
  100101                    000111
  100110                    000111
  100111                    111000
  101000                    000111
```

```
  101001                    000111
  101010                    000111
  101011                    000111
  101100                    000111
  101101                    000111
  101110                    000111
  101111                    000111
  110000                    000111
  110001                    000111
  110010                    000111
  110011                    000111
  110100                    000111
  110101                    000111
  110110                    000111
  110111                    000111
  111000                    000111
  111001                    000111
  111010                    000111
  111011                    000111
  111100                    000111
  111101                    000111
  111110                    000001
  111111                    000000
```

show class-of-service translation-table name exp-trans-table

```
user@host> show class-of-service translation-table name exp-trans-table
Translation Table: exp-trans-table, Translation table type: exp-to-exp, Index:
9048
  From Code point    To Code Point
  000                          101
  001                          111
  010                          101
  011                          111
  100                          101
  101                          101
  110                          001
  111                          000
```

show class-of-service translation-table type to-dscp-ipv6-from-dscp-ipv6

```
user@host> show class-of-service translation-table type to-dscp-ipv6-from-dscp-ipv6
Translation Table: dscp-ipv6-trans-table, Translation table type:
dscp-ipv6-to-dscp-ipv6, Index: 64704
  From Code point    To Code Point
  000000                    000111
  000001                    000111
  000010                    000111
  000011                    000111
  000100                    000111
  000101                    000111
  000110                    000111
  000111                    111000
  001000                    000111
  001001                    000111
  001010                    000111
  001011                    000111
  001100                    000111
  001101                    000111
  001110                    000111
```

001111	000111
010000	000111
010001	000111
010010	000111
010011	000111
010100	000111
010101	000111
010110	000111
010111	000111
011000	000111
011001	000111
011010	000111
011011	000111
011100	000111
011101	000111
011110	000111
011111	000111
100000	000111
100001	000111
100010	000111
100011	000111
100100	000111
100101	000111
100110	000111
100111	111000
101000	000111
101001	000111
101010	000111
101011	000111
101100	000111
101101	000111
101110	000111
101111	000111
110000	000111
110001	000111
110010	000111
110011	000111
110100	000111
110101	000111
110110	000111
110111	000111
111000	000111
111001	000111
111010	000111
111011	000111
111100	000111
111101	000111
111110	000001
111111	000000

PART 4

Index

- Index on page 103

Index

Symbols

#, comments in configuration statements....xiv
( ), in syntax descriptions....xiv
< >, in syntax descriptions....xiv
[ ], in configuration statements....xiv
{ }, in configuration statements....xiv
| (pipe), in syntax descriptions....xiv

B

BA classifiers
    bridging....31
    inner VLAN tag....31
braces, in configuration statements....xiv
brackets
    angle, in syntax descriptions....xiv
    square, in configuration statements....xiv
bridging
    BA classifiers....31

C

classification
    behavior aggregate....3
        applying DSCP IPv6 to an interface....57
        applying MPLS EXP to routing instances....59
        applying to an interface....24
        default forwarding classes and loss priorities....6
        defining custom....21
        example configuration....61
        global classifiers....60
        types of BA classifiers....5
        wildcard routing instances....60
    for IEEE 802.1ad traffic....65
classification and rewrite
    at global and physical interface levels, example configuration....53
    at global and physical interface levels, overview....8
classifiers statement
    usage guidelines....5, 59
code-point aliases....9
code-point-aliases statement....73
    usage guidelines....9
code-points statement....73
    usage guidelines....5
comments, in configuration statements....xiv
conventions
    text and syntax....xiii
CoS
    for IEEE 802.1ad....65
    mapping, displaying
        code point aliases to bit patterns....94
        translation table bits....96
curly braces, in configuration statements....xiv
customer support....xv
    contacting JTAC....xv

D

DiffServ....9
documentation
    comments on....xv
DSCP....9
    for VPLS....7
    for VPLS, example....51
DSCP IPv6
    rewrites and forwarding class maps....52
dscp statement....76
    usage guidelines....6, 21
dscp-ipv6 statement....77
    usage guidelines....6, 21

E

exp statement....78
    usage guidelines....6, 21
explicit-null statement
    with MPLS EXP classifiers....63

F

font conventions....xiii
forwarding class maps
    DSCP IPv6 rewrites....52

I

IEEE 802.1ad
    CoS classification of traffic....65
ieee-802.1 statement
    classifier on physical interface....79
    rewrite rules on logical interface....80
ieee-802.1ad statement....81

import statement
usage guidelines....5
inet-precedence statement....82
usage guidelines....6, 21
interfaces
CoS classifiers on....5
interfaces statement
CoS....84
IPv6
DSCP rewrites and forwarding class maps....52

L
loss-priority statement
usage guidelines....5

M
manuals
comments on....xv
MPLS EXP classifiers
for explicit-null labels....63
routing instances....59
example configuration....61

P
parentheses, in syntax descriptions....xiv
PICs
ToS translation....7

R
rewrites
DSCP IPv6 and forwarding class maps....52
routing instances
MPLS EXP classifier....59
example configuration....61
routing-instance statement
usage guidelines....59
routing-instances statement....87
usage guidelines....59

S
show class-of-service code-point-aliases command....94
show class-of-service translation-table command....96
support, technical *See* technical support
syntax conventions....xiii
system-defaults statement....88

T
technical support
contacting JTAC....xv

U
unit statement
CoS....89
usage guidelines....5

V
VLAN tag
BA classifiers....31
VPLS
BA classifiers....31
with DSCP....7
with DSCP, example....51