JUNIPER NETWORKS

Junos® OS

Source Class Usage Feature Guide

Release
11.4

Published: 2011-11-14

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

Junos® OS Source Class Usage Feature Guide
Release 11.4

Revision History
November 2011—R1 Junos OS 11.4

The information in this document is current as of the date listed in the revision history.

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

List of Figures

List of Tables

PART 1

Overview

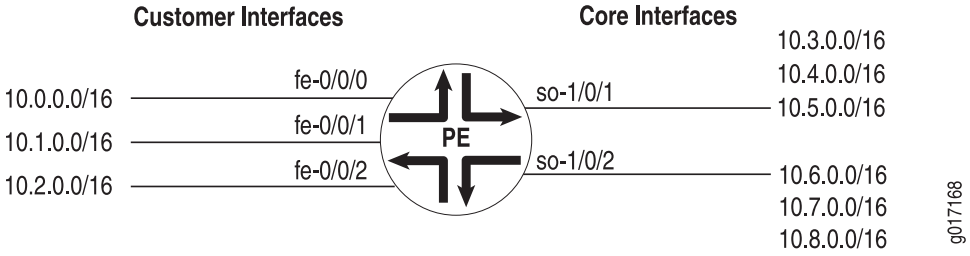
CHAPTER 1

Product Overview

Overview of Source Class Usage

Source class usage (SCU) is a logical extension of the destination class usage (DCU) concept. DCU was created so that Juniper Networks customers could count on a per-interface basis how much traffic was sent to specified prefixes. Figure 1 on page 3 shows a service provider edge (PE) router diagram.

Figure 1: DCU/SCU Concept



The Fast Ethernet interfaces contain inbound traffic from customers, and the SONET/SDH interfaces are connected to outbound public network prefixes. With DCU configured on the Fast Ethernet interfaces, you can track how much traffic is sent to a specific prefix in the core of the network originating from one of the specified interfaces (in this case, the Fast Ethernet interfaces).

However, DCU limits your ability to keep track of traffic moving in the reverse direction. It can account for all traffic that arrives on a core interface and heads toward a specific customer, but it cannot count traffic that arrives on a core interface from a specific prefix. For example, DCU can process cumulative traffic headed toward interface **fe-0/0/0**, but cannot differentiate between traffic coming only from **10.3.0.0/16** and traffic coming from all prefixes.

You can track source-based traffic by using SCU, which allows you to monitor the amount of traffic originating from a specific prefix. With this feature, usage can be tracked and customers can be billed for the traffic they receive.

Related Documentation

- Source Class Usage
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU Configuration on page 25
- Example: SCU with Layer 3 VPNs Configuration on page 33

Guidelines for Configuring SCU

When you enable SCU or DCU, keep the following information in mind:

- In Junos OS Release 5.6 and later for M Series routers only, you can use a source class or a destination class as a match condition in a firewall filter. To configure, include the **destination-class** or **source-class** statement at the **[edit firewall filter *firewall-name* term *term-name* from]** hierarchy level. For more information about firewall filters, see the *Junos Policy Framework Configuration Guide*.
- You can assign up to 126 source classes and 126 destination classes.
- A source or destination class is applied to a packet only once during the routing table lookup. When a network prefix matches a class-usage policy, SCU is assigned to packets first; DCU is assigned only if SCU has not been assigned. Be careful when using both class types, since misconfiguration can result in uncounted packets. The following example explores one potential mishap:

  A packet arrives on a router interface configured for both SCU and DCU. The packet's source address matches an SCU class, and its destination matches a DCU class. Consequently, the packet is subjected to a source lookup and is marked with the SCU class. The DCU class is ignored. As a result, the packet is forwarded to the outbound interface with only the SCU class still intact.

  However, the outbound interface lacks an SCU configuration. When the packet is ready to leave the router, the router detects that the output interface is not configured for SCU and the packet is not counted by SCU. Likewise, even though the prefix matched the DCU prefix, the DCU counters do not increment because DCU was superseded by SCU at the inbound interface.

To solve this problem, make sure you configure both the inbound and outbound interfaces completely or configure only one class type per interface per direction.

- Classes cannot be mapped to directly connected prefixes configured on local interfaces. This is true for DCU and SCU classes.
- If you use multiple terms within a single policy, you only need to configure the policy name and apply it to the forwarding table once. This makes it easier to change options within your terms without having to reconfigure the main policy.

- Execute command line interface (CLI) **show** commands and accounting profiles at the desired outbound interface to track SCU traffic. SCU counters increment at the SCU **output** interface.
- Apply your classes to the inbound and outbound interfaces by means of the **input** and **output** SCU interface parameters.
- On M320 and T Series routers, the source and destination classes are not carried across the platform fabric. For these routers, SCU and DCU accounting is performed before the packet enters the fabric and DCU is performed before output filters are evaluated.
- If an output filter drops traffic on M Series routers other than the M120 router and M320 router, the dropped packets are excluded from DCU statistics. If an output filter drops traffic on M320 and T Series routers, the dropped packets are included in DCU statistics.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13
- Example: SCU Configuration on page 25

CHAPTER 2

System Requirements

System Requirements for SCU

To implement SCU, your system must meet these requirements:

- Junos OS Release 8.2 or later for M120 and MX Series router support
- Junos OS Release 6.2 or later for IPv6 SCU
- Junos OS Release 5.6 or later to use a source class or a destination class as a match condition in a firewall filter
- Junos OS Release 5.4 or later for IPv4 SCU
- Three Juniper Networks M Series, MX Series, or T Series routers for basic SCU and five routers for SCU with Layer 3 VPNs. One router acts as a source class usage transit router, and the other routers are used to generate traffic or participate in the Layer 3 VPN.
- For M Series and T Series routers, a Tunnel Services PIC for SCU with Layer 3 VPNs

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- Roadmap for Configuring SCU on page 13
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU Configuration on page 25
- Example: SCU with Layer 3 VPNs Configuration on page 33

CHAPTER 3

Glossary

Terms and Acronyms for SCU

D

destination address (DA) — The IP address of a device intended as the receiver for a packet. This address is included in the IP header and is the main address analyzed by the router during routing table lookups and DCU.

destination class usage (DCU) — A method of grouping certain types of traffic and monitoring these groups through CLI **show** commands, accounting profiles, or SNMP. DCU uses a destination address lookup when determining group membership. For more information about DCU, see the *Junos Policy Framework Configuration Guide*.

S

source address (SA) — The IP address of a device sending a packet. This address is included in the IP header and is analyzed by the router for a variety of services, including source-based filtering, policing, class of service (CoS), and SCU.

source class usage (SCU) — A method of grouping certain types of traffic and monitoring these groups through CLI **show** commands, accounting profiles, or SNMP. SCU uses a source address lookup when determining group membership. For more information about SCU, see the *Junos Policy Framework Configuration Guide*.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- Roadmap for Configuring SCU on page 13
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13

PART 2

Configuration

CHAPTER 4

Configuration Overview

Roadmap for Configuring SCU

To configure source class usage (SCU), you must:

1. Create a routing policy that includes prefix route filters that indicate the IPv4 or IPv6 source addresses to monitor. See "Configuring Route Filters and Source Classes in a Routing Policy" on page 15.
2. Apply the filters to the forwarding table. See "Applying the Policy to the Forwarding Table" on page 16.
3. Enable accounting on the inbound and outbound interfaces. See "Enabling Accounting on Inbound and Outbound Interfaces" on page 16.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Example: SCU Configuration on page 25

Roadmap for Configuring SCU with Layer 3 VPNs

SCU can be implemented over regular interfaces; it is also used in combination with Layer 3 VPNs. When you view SCU traffic on an ingress provider edge (PE) router, use the standard procedure outlined in "Roadmap for Configuring SCU" on page 13. However, when you enable packet counting for Layer 3 VPNs at the egress point of the MPLS tunnel, you need to take some additional steps, as follows:

1. Configure SCU on the virtual loopback tunnel (vt) interface of the egress PE router. See "Configuring Input SCU on the vt Interface of the Egress PE Router" on page 19.
2. Map the SCU-enabled input interface of that router to the virtual routing and forwarding (VRF) instance. See "Mapping the SCU-Enabled vt Interface to the VRF Instance" on page 20.

3. Configure SCU on the output interface of the egress router. See "Configuring SCU on the Output Interface" on page 20.
4. Configure an accounting profile and associate the source class with that accounting profile. You can also specify the filename for the data capture, a class usage profile name, and an interval indicating how often you want the SCU information to be saved. See "Associating an Accounting Profile with SCU Classes" on page 21.

NOTE: **SCU is not supported over Layer 2 VPNs.**

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Example: SCU with Layer 3 VPNs Configuration on page 33

CHAPTER 5

Basic Configuration Steps

Configuring Route Filters and Source Classes in a Routing Policy

Begin configuring SCU by creating prefix route filters in a policy statement. These prefixes indicate the IPv4 or IPv6 source addresses to monitor. Within the policy statement, you must define and name the source classes attached to the filters.

```
[edit policy-options]
policy-statement policy-name {
   term term-name {
      from {
         route-filter address/prefix;
      }
      then source-class class-name;
   }
}
```

An alternate configuration method, using the **forwarding-class** policy action, is even more flexible. It allows your IPv4 or IPv6 route filters to apply to an SCU profile, a DCU profile, or both simultaneously. Additionally, if you have only one term, you can implement the **from** and **then** statements at the **[edit policy-options policy-statement *policy-name*]** hierarchy level.

```
[edit policy-options]
policy-statement policy-name {
   from {
      route-filter 105.15.0.0/16 orlonger;
   }
   then forwarding-class class-name;
}
```

A third option is the existing DCU parameter of **destination-class**. For more information on DCU, see the *Junos Policy Framework Configuration Guide*.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3

- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13
- Example: SCU Configuration on page 25

Applying the Policy to the Forwarding Table

Next, apply the policy you created to the forwarding table. When you apply the policy, the network prefixes you defined are marked with the appropriate source class.

```
[edit routing-options]
forwarding-table {
   export policy-name;
}
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13
- Example: SCU Configuration on page 25

Enabling Accounting on Inbound and Outbound Interfaces

Unlike DCU, which only requires implementation on a single interface, accounting for SCU must be enabled on two interfaces: the inbound and outbound physical or logical interfaces traversed by the source class. You must define explicitly the two interfaces on which SCU monitored traffic is expected to arrive and depart. This is because SCU performs two lookups in the routing table: a source address (SA) and a destination address (DA) lookup. In contrast, DCU only has a single destination address lookup. By specifying the addresses involved in the additional SCU SA lookup, you minimize the performance impact on your router.

An individual SCU interface can be configured as an input interface, an output interface, or both. SCU can be enabled in an IPv4 (**family inet**) or IPv6 (**family inet6**) network. To configure SCU accounting, include the **source-class-usage** statement at the **[edit interfaces** *interface-name* **unit** *logical-unit-number* **family (inet | inet 6) accounting]** hierarchy level:

```
[edit]
interfaces {
   interface-name {
      unit unit-number {
         family (inet | inet6) {
            accounting {
               source-class-usage {
                  (input | output | input output);
               }
               destination-class-usage;
            }
```

```
        }
      }
    }
  }
```

After the full SCU configuration is enabled, every packet arriving on an SCU input interface is subjected to an SA-based lookup and then a DA-based lookup. In addition, an individual set of counters for every configured SCU class is maintained by the router on a per-interface and per-protocol family basis.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13
- Example: SCU Configuration on page 25

CHAPTER 6

Configuring SCU with Layer 3 VPNs

Configuring Input SCU on the vt Interface of the Egress PE Router

To enable SCU in a Layer 3 VPN, configure source class usage on the virtual loopback tunnel (**vt**) interface of the egress PE router that is either configured for or equipped with a Tunnel PIC. The interface is equivalent to the inbound SCU interface, so use the **input** statement at the **[edit interfaces *vt-interface-number* unit 0 family inet accounting source-class-usage]** hierarchy level:

```
[edit]
interfaces {
   vt-0/3/0 {
      unit 0 {
         family inet {
            accounting {
               source-class-usage {
                  input;
               }
            }
         }
      }
   }
}
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU with Layer 3 VPNs Configuration on page 33

Mapping the SCU-Enabled vt Interface to the VRF Instance

Next, include the VPN loopback tunnel interface in the desired VRF instance at the **[edit routing-instances *routing-instance-name*]** hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name  {
    instance-type vrf;
    interface at-2/1/1.0;
    interface vt-0/3/0.0;
    route-distinguisher 10.250.14.225:100;
    vrf-import import-policy-name;
    vrf-export export-policy-name;
    protocols {
      bgp {
        group to-r4 {
          local-address 10.20.253.1;
          peer-as 400;
          neighbor 10.20.253.2;
        }
      }
    }
  }
}
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU with Layer 3 VPNs Configuration on page 33

Configuring SCU on the Output Interface

Since VPN traffic enters the egress router through the VPN loopback tunnel interface, you still need to determine the exit interface for this traffic. To complete your SCU configuration, configure the output version of source class usage on the exit interface of your egress router:

```
[edit interfaces]
at-1/1/0 {
  unit 0 {
    family inet {
      accounting {
        source-class-usage {
          output;
        }
      }
    }
  }
```

```
}
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU with Layer 3 VPNs Configuration on page 33

Associating an Accounting Profile with SCU Classes

Once your source classes are defined, implemented on the inbound and outbound interfaces, and applied to the forwarding table, you are ready to associate the source class with an accounting profile. Configure the accounting profile at the **[edit accounting-options class-usage-profile]** hierarchy level. You can associate either an SCU source class or a DCU destination class with the accounting profile. You can also specify the filename for the data capture, a class usage profile name, and an interval (in minutes) indicating how often you want the SCU information to be saved to the file.

```
[edit]
accounting-options {
   file filename;
   class-usage-profile profile-name {
      file filename;
      interval minutes;
      source-classes {
         source-class-name;
      }
      destination-classes {
         destination-class-name;
      }
   }
}
```

NOTE: SCU accounting occurs on the outbound interface before output filter processing. If an SCU-marked packet is discarded in the router, the SCU counters can indicate more traffic than actually exists. You must use filter counters or traceoptions logs to ensure that all packets dropped by the SCU filter are recorded. If logged, you can subtract the discarded packets from the SCU counter tallies and calculate the true traffic profile.

Because DCU accounting occurs after the filtering process, DCU is unaffected by this disclaimer.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7

- Roadmap for Configuring SCU with Layer 3 VPNs on page 13
- Example: SCU with Layer 3 VPNs Configuration on page 33

Verifying Your SCU Accounting Profile

Purpose To view the results of the SCU accounting profile you created.

Action Navigate to the **/var/log** directory of your router. It should contain the designated class usage profile log. The layout of an SCU profile looks like this:

```
profile_name,epoch-timestamp,interface-name,source-class-name,packet-count,
byte-count
```

An example of the actual output from a profile looks like this:

```
scu_profile,980313078,ge-1/0/0.0,gold,82,6888
scu_profile,980313078,ge-1/0/0.0,silver,164,13776
scu_profile,980313078,ge-1/0/0.0,bronze,0,0
scu_profile,980313678,ge-1/0/0.0,gold,82,6888
scu_profile,980313678,ge-1/0/0.0,silver,246,20664
scu_profile,980313678,ge-1/0/0.0,bronze,0,0
```

To view the parameters of your SCU accounting profile, you can use the **show accounting-options class-usage-profile** ***scu-profile-name*** command.

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Associating an Accounting Profile with SCU Classes on page 21

PART 3

Examples

CHAPTER 7

Examples

Example: SCU Configuration

Example: SCU Configuration

Figure 2: SCU Topology Diagram

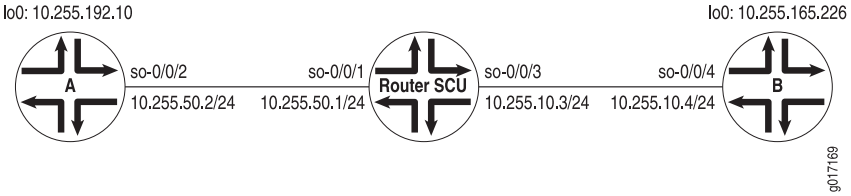


Figure 2 on page 25 shows a basic SCU configuration with three routers. Source routers A and B use loopback addresses as the prefixes to be monitored. Most of the configuration tasks and actual monitoring occurs on transit Router SCU.

Begin your configuration on Router A. The loopback address on Router A contains the origin of the prefix that is to be assigned to source class A on Router SCU. However, no SCU processing happens on this router. Therefore, configure Router A for basic OSPF routing and include your loopback interface and interface **so-0/0/2** in the OSPF process.

Router A:

```
[edit]
interfaces {
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.255.50.2/24;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
```

```
            address 10.255.192.10/32;
          }
        }
      }
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-0/0/2.0;
          interface lo0.0;
        }
      }
    }
```

Router SCU handles the bulk of the activity in this example. On Router SCU, enable source class usage on the inbound and outbound interfaces at the **[edit interfaces *interface-name* unit *unit-number* family inet accounting]** hierarchy level. Make sure you specify the expected traffic: input, output, or, in this case, both.

Next, configure a route filter policy statement that matches the prefixes of the loopback addresses from routers A and B. Include statements in the policy that classify packets from Router A in one group named **scu-class-a** and packets from Router B in a second class named **scu-class-b**. Notice the efficient use of a single policy containing multiple terms.

Last, apply the policy to the forwarding table.

Router SCU

```
[edit]
interfaces {
  so-0/0/1 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            input;
            output;
          }
        }
        address 10.255.50.1/24;
      }
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            input;
            output;
          }
        }
        address 10.255.10.3/24;
      }
    }
  }
```

```
    lo0 {
      unit 0 {
        family inet {
          address 10.255.6.111/32;
        }
      }
    }
}
protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/1.0;
        interface so-0/0/3.0;
      }
    }
}
routing-options {
    forwarding-table {
      export scu-policy;
    }
}
policy-options {
    policy-statement scu-policy {
      term 0 {
        from {
          route-filter 10.255.192.0/24 orlonger;
        }
        then source-class scu-class-a;
      }
      term 1 {
        from {
          route-filter 10.255.165.0/24 orlonger;
        }
        then source-class scu-class-b;
      }
    }
}
```

Complete the configuration tasks on Router B. Just as Router A provides a source prefix, Router B's loopback address matches the prefix assigned to **scu-class-b** on Router SCU. Again, no SCU processing happens on this router, so configure Router B for basic OSPF routing and include your loopback interface and interface **so-0/0/4** in the OSPF process.

Router B:

```
[edit]
interfaces {
    so-0/0/4 {
      unit 0 {
        family inet {
          address 10.255.10.4/24;
        }
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.165.226/32;
```

```
        }
      }
    }
  }
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/4.0;
        interface lo0.0;
      }
    }
  }
```

Verifying Your Work

To verify that SCU is functioning properly, use the following commands:

- **show interfaces** *interface-name* **statistics**
- **show interfaces** *interface-name* (**extensive** | **detail**)
- **show route** (**extensive** | **detail**)
- **show interfaces source-class** *source-class-name interface-name*
- **clear interface** *interface-name* **statistics**

You should always verify SCU statistics at the outbound SCU interface on which you configured the **output** statement. You can perform the following three steps to check the functionality of SCU:

1. Clear all counters on your SCU-enabled router and verify that they are empty.
2. Send a ping from one edge router to another edge router to generate SCU traffic across the SCU-enabled router.
3. Verify that the counters are incrementing correctly on the outbound interface.

The following section shows the output of these commands as used with the configuration example.

```
user@scu> clear interfaces statistics all

user@scu>  show interfaces so-0/0/1.0 statistics
  Logical interface so-0/0/1.0 (Index 4) (SNMP ifIndex 119)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                                Packets                Bytes
                        scu-class-a        0        0
        scu-class-b        0        0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.255.50/24, Local: 10.255.50.1

user@scu> show interfaces so-0/0/3.0 statistics
  Logical interface so-0/0/3.0 (Index 6) (SNMP ifIndex 113)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                                Packets                Bytes
```

```
                    scu-class-a                    0                    0
                    scu-class-b                    0                    0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.255.10/24, Local: 10.255.10.3

user@scu> show interfaces source-class scu-class-a so-0/0/3.0
    Protocol inet
      Source class                              Packets                Bytes
                    scu-class-a                    0                    0

user@scu> show interfaces source-class scu-class-b so-0/0/1.0
    Protocol inet
      Source class                              Packets                Bytes
                    scu-class-b                    0                    0

user@routerB> ping 10.255.192.10 source 10.255.165.226 rapid 10000

user@routerA> ping 10.255.165.226 source 10.255.192.10 rapid 10000

user@scu> show interfaces source-class scu-class-a so-0/0/3.0
    Protocol inet
      Source class                              Packets                Bytes
                    scu-class-a                  20000              1680000

user@scu> show interfaces source-class scu-class-a so-0/0/1.0
    Protocol inet
      Source class                              Packets                Bytes
                    scu-class-b                  20000              1680000

user@scu> show interfaces so-0/0/3.0 statistics
  Logical interface so-0/0/3.0 (Index 6) (SNMP ifIndex 113)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                              Packets                Bytes
                    scu-class-a      20000     1680000
                    scu-class-b                    0                    0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.255.10/24, Local: 10.255.10.3

user@scu> show interfaces so-0/0/1.0 statistics
  Logical interface so-0/0/1.0 (Index 4) (SNMP ifIndex 119)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                              Packets                Bytes
                    scu-class-a                    0                    0
                    scu-class-b      20000     1680000
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.255.50/24, Local: 10.255.50.1


user@scu> show route extensive 10.255.192.0

inet.0: 26 destinations, 28 routes (25 active, 0 holddown, 1 hidden)
10.255.192.0/18 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.255.192.0/18 -> {so-0/0/1.0}
Source class: scu-class-a
        *OSPF   Preference: 150
                Next hop: via so-0/0/1.0, selected
                State: <Active Int Ext>
```

```
                Age: 2:49:31  Metric: 0  Tag: 0
                Task: OSPF
                Announcement bits (1): 0-KRT
                AS path: I


user@scu> show route extensive 10.255.165.0
inet.0: 26 destinations, 28 routes (25 active, 0 holddown, 1 hidden)
10.255.165.0/20 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.255.165.0/20 -> {so-0/0/3.0}
Source class: scu-class-b
        *OSPF   Preference: 150
                Next hop: via so-0/0/3.0, selected
                State: <Active Int Ext>
                Age: 2:49:31  Metric: 0  Tag: 0
                Task: OSPF
                Announcement bits (1): 0-KRT
                AS path: I


user@scu> show interfaces so-0/0/1 detail
Physical interface: so-0/0/1, Enabled, Physical link is Up
  Interface index: 12, SNMP ifIndex: 17, Generation: 11
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 46 (last seen 00:00:01 ago)
    Output: 45 (last sent 00:00:00 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Not-configured
  Last flapped   : 2002-04-19 11:49:22 PDT (03:10:09 ago)
  Statistics last cleared: 2002-04-19 14:52:04 PDT (00:07:27 ago)
  Traffic statistics:
   Input  bytes  :              1689276                 40 bps
   Output bytes  :              1689747                 48 bps
   Input  packets:                20197                  0 pps
   Output packets:                20200                  0 pps
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                20053                20053                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                 146                  146                    0

  SONET alarms   : None
  SONET defects  : None
  Logical interface so-0/0/1.0 (Index 4) (SNMP ifIndex 119) (Generation 3)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
```

```
      Flags: SCU-in, SCU-out
    Generation: 6 Route table: 0
       Source class              Packets        Bytes
         scu-class-a         0         0
         scu-class-b       20000      1680000
      Filters: Input: icmp-so-0/0/1.0-i, Output: icmp-so-0/0/1.0-o
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.255.50/24, Local: 10.255.50.1, Broadcast: Unspecified,
        Generation: 8

user@scu>  show interfaces so-0/0/1 extensive
Physical interface: so-0/0/1, Enabled, Physical link is Up
  Interface index: 12, SNMP ifIndex: 17, Generation: 11
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 51 (last seen 00:00:04 ago)
    Output: 50 (last sent 00:00:05 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Not-configured
  Last flapped   : 2002-04-19 11:49:22 PDT (03:11:05 ago)
  Statistics last cleared: 2002-04-19 14:52:04 PDT (00:08:23 ago)
  Traffic statistics:
   Input  bytes  :              1689884                 264 bps
   Output bytes  :              1690388                 280 bps
   Input  packets:                20215                   0 pps
   Output packets:                20217                   0 pps
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0,
    Bucket drops: 0, Policed discards: 0, L3 incompletes: 0,
    L2 channel errors: 0, L2 mismatch timeouts: 0, HS link CRC errors: 0,
    HS link FIFO overflows: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Aged packets: 0,
    HS link FIFO underflows: 0
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort               20053               20053                    0

    1 expedited-fo                  0                   0                    0

    2 assured-forw                  0                   0                    0

    3 network-cont                164                 164                    0

  SONET alarms   : None
  SONET defects  : None
  SONET PHY:            Seconds        Count  State
    PLL Lock                  0            0  OK
    PHY Light                 0            0  OK
  SONET section:
    BIP-B1                    0            0
    SEF                       0            0  OK
    LOS                       0            0  OK
```

```
  LOF                       0            0  OK
  ES-S                      0
  SES-S                     0
  SEFS-S                    0
SONET line:
  BIP-B2                    0            0
  REI-L                     0            0
  RDI-L                     0            0  OK
  AIS-L                     0            0  OK
  BERR-SF                   0            0  OK
  BERR-SD                   0            0  OK
  ES-L                      0
  SES-L                     0
  UAS-L                     0
  ES-LFE                    0
  SES-LFE                   0
  UAS-LFE                   0
SONET path:
  BIP-B3                    0            0
  REI-P                     0            0
  LOP-P                     0            0  OK
  AIS-P                     0            0  OK
  RDI-P                     0            0  OK
  UNEQ-P                    0            0  OK
  PLM-P                     0            0  OK
  ES-P                      0
  SES-P                     0
  UAS-P                     0
  ES-PFE                    0
  SES-PFE                   0
  UAS-PFE                   0
Received SONET overhead:
  F1      : 0x00, J0      : 0x00, K1      : 0x00, K2      : 0x00
  S1      : 0x00, C2      : 0xcf, C2(cmp) : 0xcf, F2      : 0x00
  Z3      : 0x00, Z4      : 0x00, S1(cmp) : 0x00, V5      : 0x00
  V5(cmp) : 0x00
Transmitted SONET overhead:
  F1      : 0x00, J0      : 0x01, K1      : 0x00, K2      : 0x00
  S1      : 0x00, C2      : 0xcf, F2      : 0x00, Z3      : 0x00
  Z4      : 0x00, V5      : 0x00
Received path trace: e so-0/0/1
  65 20 73 6f 2d 30 2f 30 2f 31 00 00 00 00 00 00   e so-0/0/1......
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 0d 0a   ................
Transmitted path trace: scu so-0/0/1
  67 68 62 20 73 6f 2d 30 2f 30 2f 31 00 00 00 00   scu so-0/0/1....
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
HDLC configuration:
  Policing bucket: Disabled
  Shaping bucket : Disabled
  Giant threshold: 4484, Runt threshold: 3
Packet Forwarding Engine configuration:
  Destination slot: 0, PLP byte: 1 (0x00)
  CoS transmit queue          Bandwidth          Buffer     Priority   Limit
                            %          bps   %        bytes
  0 best-effort             0            0   0            0      low    none
  1 expedited-forwarding    0            0   0            0      low    none
  2 assured-forwarding      0            0   0            0      low    none
```

```
  3 network-control         0            0    0            0      low    none
Logical interface so-0/0/1.0 (Index 4) (SNMP ifIndex 119) (Generation 3)
  Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
  Protocol inet, MTU: 4470
    Flags: SCU-in, SCU-out
  Generation: 6 Route table: 0
     Source class            Packets        Bytes
       scu-class-a                 0            0
       scu-class-b             20000      1680000
    Filters: Input: icmp-so-0/0/1.0-i, Output: icmp-so-0/0/1.0-o
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 10.255.50/24, Local: 10.255.50.1, Broadcast: Unspecified,
      Generation: 8
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7
- Roadmap for Configuring SCU on page 13

Example: SCU with Layer 3 VPNs Configuration

- Example: SCU in a Layer 3 VPN Configuration on page 33
- Verifying Your Work on page 40

Example: SCU in a Layer 3 VPN Configuration

Figure 3: SCU in a Layer 3 VPN Topology Diagram

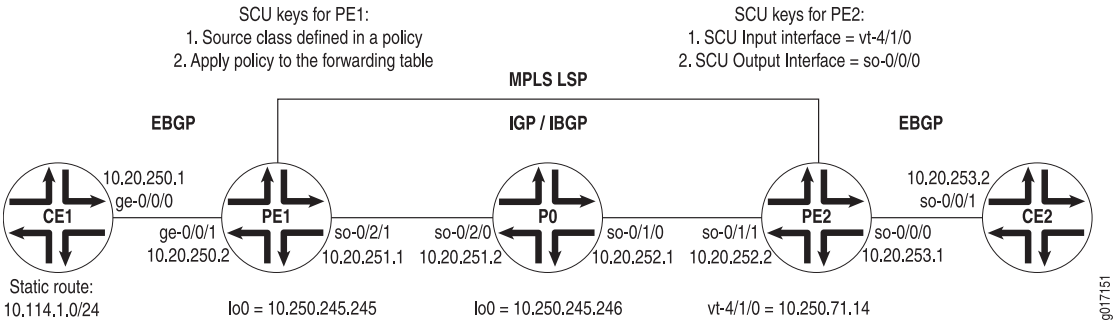


Figure 3 on page 33 displays a Layer 3 VPN topology. CE1 and CE2 are customer edge (CE) routers connected by a VPN through provider routers PE1, P0, and PE2. EBGP is established between routers CE1 and PE1, IBGP connects routers PE1 and PE2 over an IS-IS/MPLS/LDP core, and a second EBGP connection flows between routers PE2 and CE2.

On Router CE1, begin your VPN by setting up an EBGP connection to PE1. Install a static route of **10.114.1.0/24** and advertise this route to your EBGP neighbor.

Router CE1

```
[edit]
interfaces {
  ge-0/0/0 {
    unit 0 {
```

```
        family inet {
          address 10.20.250.1/30;
        }
      }
    }
  }
  routing-options {
    static {
      route 10.114.1.0/24 reject;
    }
    autonomous-system 100;
  }
  protocols {
    bgp {
      group to-pe1 {
        local-address 10.20.250.1;
        export inject-direct;
        peer-as 300;
        neighbor 10.20.250.2;
      }
    }
  }
  policy-options {
    policy-statement inject-direct {
      term 1 {
        from {
          protocol static;
          route-filter 10.114.1.0/24 exact;
        }
        then accept;
      }
      term 2 {
        from protocol direct;
        then accept;
      }
    }
  }
```

On PE1, complete the EBGP connection to CE1 through a VRF routing instance. Set an export policy for your VRF instance that puts BGP traffic into a community, and an import policy that accepts like community traffic from your VPN neighbor. Lastly, configure an IBGP relationship to Router PE2 that runs over an IS-IS, MPLS, and LDP core.

Router PE1

```
[edit]
interfaces {
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 10.20.250.2/30;
      }
    }
  }
  so-0/2/1 {
    unit 0 {
      family inet {
        address 10.20.251.1/30;
```

```
        }
        family iso;
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.250.245.245/32;
        }
        family iso;
        family mpls;
      }
    }
}
routing-options {
    autonomous-system 300;
}
protocols {
    mpls {
      interface so-0/2/1;
    }
    bgp {
      group ibgp {
        type internal;
        local-address 10.250.245.245;
        family inet-vpn {
          unicast;
        }
        neighbor 10.250.71.14;
      }
    }
    isis {
      interface so-0/2/1;
    }
    ldp {
      interface so-0/2/1;
    }
}
policy-options {
    policy-statement red-import {
      from {
        protocol bgp;
        community red-com;
      }
      then accept;
    }
    policy-statement red-export {
      from protocol bgp;
      then {
        community add red-com;
        accept;
      }
    }
    community red-com members target:20:20;
}
```

```
routing-instances {
  red {
    instance-type vrf;
    interface ge-0/0/1.0;
    route-distinguisher 10.250.245.245:100;
    vrf-import red-import;
    vrf-export red-export;
    protocols {
      bgp {
        group to-ce1 {
          local-address 10.20.250.2;
          peer-as 100;
          neighbor 10.20.250.1;
        }
      }
    }
  }
}
```

On P0, connect the IBGP neighbors located at PE1 and PE2. Remember to include VPN-related protocols (MPLS, LDP, and IGP) on all interfaces.

Router P0

```
[edit]
interfaces {
  so-0/1/0 {
    unit 0 {
      family inet {
        address 10.20.252.1/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/2/0 {
    unit 0 {
      family inet {
        address 10.20.251.2/30;
      }
      family iso;
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.250.245.246/32;
      }
      family iso;
      family mpls;
    }
  }
}
routing-options {
  autonomous-system 300;
}
protocols {
```

```
    mpls {
      interface so-0/1/0;
      interface so-0/2/0;
    }
    isis {
      interface all;
    }
    ldp {
      interface all;
    }
  }
```

On PE2, complete the IBGP relationship to Router PE1. Establish an EBGP connection to CE2 through a VRF routing instance. Set an export policy for the VRF instance that places BGP traffic into a community, and an import policy that accepts like community traffic from the VPN neighbor. Next, establish a policy that adds the static route from CE1 to a source class called **GOLD1**. Also, export this SCU policy into the forwarding table. Finally, set your **vt** interface as the SCU input interface and establish the CE-facing interface **so-0/0/0** as the SCU output interface.

Router PE2

```
[edit]
interfaces {
  so-0/1/1 {
    unit 0 {
      family inet {
        address 10.20.252.2/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/0 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            output;
          }
        }
        address 10.20.253.1/30;
      }
    }
  }
  vt-4/1/0 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            input;
          }
        }
        address 10.250.71.14/32;
      }
      family iso;
      family mpls;
```

```
    }
  }
}
routing-options {
  autonomous-system 300;
  forwarding-table {
    export inject-customer2-dest-class;
  }
}
protocols {
  mpls {
    interface so-0/1/1;
    interface vt-4/1/0;
  }
  bgp {
    group ibgp {
      type internal;
      local-address 10.250.71.14;
      family inet-vpn {
        unicast;
      }
      neighbor 10.250.245.245;
    }
  }
  isis {
    interface so-0/1/1;
  }
  ldp {
    interface so-0/1/1;
  }
}
routing-instances {
  red {
    instance-type vrf;
    interface so-0/0/0.0;
    interface vt-4/1/0.0;
    route-distinguisher 10.250.71.14:100;
    vrf-import red-import;
    vrf-export red-export;
    protocols {
      bgp {
        group to-ce2 {
          local-address 10.20.253.1;
          peer-as 400;
          neighbor 10.20.253.2;
        }
      }
    }
  }
}
policy-options {
  policy-statement red-import {
    from {
      protocol bgp;
      community red-com;
    }
```

```
        then accept;
    }
    policy-statement red-export {
        from protocol bgp;
        then {
            community add red-com;
            accept;
        }
    }
    policy-statement inject-customer2-dest-class {
        term term-gold1-traffic {
            from {
                route-filter 10.114.1.0/24 exact;
            }
            then source-class GOLD1;
        }
    }
    community red-com members target:20:20;
}
```

On Router CE2, complete the VPN path by finishing the EBGP connection to PE2.

Router CE2

```
[edit]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.20.253.2/30;
            }
        }
    }
}
routing-options {
    autonomous-system 400;
}
protocols {
    bgp {
        group to-pe2 {
            local-address 10.20.253.2;
            export inject-direct;
            peer-as 300;
            neighbor 10.20.253.1;
        }
    }
}
policy-options {
    policy-statement inject-direct {
        from {
            protocol direct;
        }
        then accept;
    }
}
```

Verifying Your Work

To verify that SCU is functioning properly in the Layer 3 VPN, use the following commands:

- **show interfaces** *interface-name* **statistics**
- **show interfaces source-class** *source-class-name interface-name*
- **show interfaces** *interface-name* (**extensive** | **detail**)
- **show route** (**extensive** | **detail**)
- **clear interface** *interface-name* **statistics**

You should always verify SCU statistics at the outbound SCU interface on which you configured the **output** statement. To check SCU functionality, follow these steps:

1. Clear all counters on your SCU-enabled router and verify they are empty.
2. Send a ping from the ingress CE router to the second CE router to generate SCU traffic across the SCU-enabled VPN route.
3. Verify that the counters are incrementing correctly on the outbound interface.

The following section shows the output of these commands used with the configuration example.

```
user@pe2> clear interfaces statistics all

 user@pe2> show interfaces so-0/0/0.0 statistics
   Logical interface so-0/0/0.0 (Index 6) (SNMP ifIndex 113)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                                Packets                Bytes
 GOLD1                0                    0
      Addresses, Flags: Is-Preferred Is-Primary

 user@pe2> show interfaces source-class GOLD1 so-0/0/0.0
    Protocol inet
      Source class                                Packets                Bytes
GOLD1                0                    0

user@ce1> ping 10.20.253.2 source 10.114.1.1 rapid count 10000

user@scu>  show interfaces source-class GOLD1 so-0/0/0.0
    Protocol inet
      Source class                                Packets                Bytes
GOLD1               20000              1680000

user@scu>  show interfaces so-0/0/0.0 statistics
  Logical interface so-0/0/0.0 (Index 6) (SNMP ifIndex 113)
    Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
    Protocol inet, MTU: 4470
      Source class                                Packets                Bytes
GOLD1               20000              1680000
      Addresses, Flags: Is-Preferred Is-Primary
         Destination: 10.20.253/24, Local: 10.20.253.1
```

Related Documentation

- Source Class Usage
- Overview of Source Class Usage on page 3
- System Requirements for SCU on page 7

PART 4

Administration

CHAPTER 8

Commands

clear interfaces statistics

Syntax clear interfaces statistics (all | *interface-name*)

Release Information Command introduced before Junos OS Release 7.4.

Description Set interface statistics to zero. If you issue the **clear interfaces statistics** ***interface-name*** command and then perform a graceful Routing Engine switchover, the interface statistics are not cleared on the new master. Reissue the command to clear the interface statistics again.

Options all—Set statistics on all interfaces to zero.

interface-name—Set statistics on a particular interface to zero.

Required Privilege Level clear

List of Sample Output clear interfaces statistics on page 46

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear interfaces statistics

```
user@host> clear interfaces statistics
```

show accounting profile

Syntax show accounting profile *profile-name*

Release Information Command introduced before Junos OS Release 7.4.

Description Display accounting profile information.

Options *profile-name*—Name of the accounting profile.

Required Privilege Level view

List of Sample Output
show accounting profile (Interface) on page 48
show accounting profile (Filter) on page 49
show accounting profile (Destination Class) on page 49
show accounting profile (Routing Engine) on page 50

Output Fields Table 1 on page 47 lists the output fields for the **show accounting profile** command. Output fields are listed in the approximate order in which they appear.

Table 1: show accounting profile Output Fields

Field Name	Field Description
Profile	Name of the accounting profile.
Sampling interval	Configured interval, in minutes, for statistic collection.
Profile Usage Count	Number of items configured for collecting accounting statistics.
file information	Information about the accounting profile log, including: • **File**—Name of accounting profile log. If no name is explicitly provided, the name of the accounting profile is used. All statistics files are placed in the **/var/log** directory. • **maximum size**—Configured size. When the size is exceeded, the log file closes and a new log file opens. • **maximum number**—Configured maximum number of log files. • **bytes written**—Number of bytes written to the log file.
Transfer Interval	Length of time (in minutes) the file remains open, receiving statistics before it is closed, transferred, and rotated. When either the time or the file size is exceeded, the file is closed and a new one opened, whether or not a transfer site is specified.
Next Scheduled Transfer	Time at which the next transfer occurs.

Table 1: show accounting profile Output Fields *(continued)*

Field Name	Field Description
Column Labels	Names of sampled statistics. This list varies depending on the configuration: • **profile-layout**—List of data fields reported, in the order they appear in the output. • **epoch-timestamp**—Number of seconds since the epoch. • **interfaces**—(For interface, filter, and destination class profiles) Name of the interfaces on which the filter is applied. • **filter-name**—(For filter profiles) Name of the filter. • **counter-name**—(For filter profiles) Name of the counter. • **packet-count**—(For filter and destination class profiles) Number of packets for the counter. • **byte-count**—(For filter and destination class profiles) Number of bytes for the counter. • **input-bytes**—(For interface profiles) Input bytes. • **input-errors**—(For interface profiles) Generic input error packets. • **input-multicast**—(For interface profiles) Input packets arriving by multicast. • **input-packets**—(For interface profiles) Input packets. • **input-unicast**—(For interface profiles) Input unicast packets. • **output-bytes**—(For interface profiles) Output bytes. • **output-errors**—(For interface profiles) Generic output error packets. • **output-multicast**—(For interface profiles) Output packets sent by multicast. • **output-packets**—(For interface profiles) Output packets. • **output-unicast**—(For interface profiles) Output unicast packets. • **no-proto**—(For interface profiles) Packets for unsupported protocol. • **snmp-index**—(For interface profiles) SNMP index. • **destination-class-name**—(For destination class profiles) Configured destination class name. • **host name**—(For Routing Engine profiles) Hostname for the router. • **date-*yyyymmdd***—(For Routing Engine profiles) Date. • **timeofday-*hhmmss***—(For Routing Engine profiles) Time of day. • **uptime**—(For Routing Engine profiles) Time since the last reboot, in seconds. • **cpu1min**—(For Routing Engine profiles) Average system load over the last 1 minute. • **cpu5min**—(For Routing Engine profiles) Average system load over the last 5 minutes. • **cpu15min**—(For Routing Engine profiles) Average system load over the last 15 minutes.
Interface name	Name of the interface configured for this accounting profile.
Filter name	Name of the filter configured for this accounting profile.
routing-engine-stats	Routing Engine accounting profile.
Next Scheduled Collection	Time for next collection of statistics for the named interface.

Sample Output

show accounting profile (Interface)

```
user@host> show accounting profile if_prof
Profile if_prof
Sampling interval: 1 minute(s),  Profile Usage Count: 2
```

```
File accounting_profile_stats: maximum size 1048576, maximum number 5, bytes
written 2196
Transfer Interval: 15 minute(s),  Next Scheduled Transfer: 2001-06-17-18:00:45
Column Labels:
  profile-layout
  epoch-timestamp
  interface-name
  snmp-index
  input-bytes
  output-bytes
  input-packets
  output-packets
  input-unicast
  output-unicast
  input-multicast
  output-multicast
  no-proto
  input-errors
  output-errors


Interface Name              Next Scheduled Collection
fxp0.0                      2001-06-18-18:00:30
fxp0                        2001-06-18-18:01:00
```

show accounting profile (Filter)

```
user@host> show accounting profile filter_profile
Profile filter_profile
Sampling interval: 1 minute(s),  Profile Usage Count: 0
File accounting_profile_stats: maximum size 1048576, maximum number 5, bytes
written 822
Transfer Interval: 15 minute(s),  Next Scheduled Transfer: 2001-06-17-18:00:46
Column Labels:
  profile-layout
  epoch-timestamp
  interfaces
  filter-name
  counter-name
  packet-count
  byte-count


Filter Name                 Next Scheduled Collection
myfiltero                         2001-06-03-04:32:59
```

show accounting profile (Destination Class)

```
user@host> show accounting profile dcu1
Profile dcu1
Sampling interval: 1 minute(s),  Profile Usage Count: 0
File accounting_profile_stats: maximum size 1048576, maximum number 5, bytes
written 901
Transfer Interval: 15 minute(s),  Next Scheduled Transfer: 2001-06-17-18:00:46
Column Labels:
  profile-layout
  epoch-timestamp
  interface-name
  destination-class-name
  packet-count
  byte-count
```

```
Interface Name              Next Scheduled Collection
so-0/3/3                    2001-06-03-04:34:00
```

show accounting profile (Routing Engine)

```
user@host> show accounting profile rep1
Profile rep1
Sampling interval: 1 minute(s),  Profile Usage Count: 1
File accounting_profile_stats: maximum size 1048576, maximum number 5, bytes
written 901
Transfer Interval: 15 minute(s),  Next Scheduled Transfer: 2001-06-17-18:00:46
Column Labels:
  profile-layout
  epoch-timestamp
  hostname
  date-yyyymmdd
  timeofday-hhmmss
  uptime
  cpu1min
  cpu5min
  cpu15min


Interface Name              Next Scheduled Collection
routing-engine-stats        2001-06-18-18:02:31
```

show interfaces source-class

Syntax show interfaces source-class
(all | *destination-class-name logical-interface-name*)

Release Information Command introduced before Junos OS Release 7.4.
all option introduced in Junos OS Release 8.0.

Description Display information about interfaces grouped by source class.

Options all—Display information about all configured source classes.

source-class-name—Name of a logical grouping of prefixes that count packets having the source address matching those prefixes.

interface-name—Name of a logical interface.

Additional Information For interfaces that carry IPv4, IPv6, or Multiprotocol Label Switching (MPLS) traffic, you can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into sets defined as source classes and destination classes. For more information, see the *Junos OS Network Interfaces Configuration Guide*.

Required Privilege Level view

List of Sample Output show interfaces source-class all on page 52

Output Fields Table 2 on page 51 lists the output fields for the **show interfaces source-class** command. Output fields are listed in the approximate order in which they appear.

Table 2: show interfaces source-class Output Fields

Field Name	Field Description
Logical interface	Name of the logical interface.
Source class	Source class usage (SCU) counters per class for this interface.
Packets	Packets going to designated user-selected prefixes.
Bytes	Bytes going to designated user-selected prefixes.

Sample Output

show interfaces source-class all

```
user@host> show interfaces source-class all
Logical interface so-0/1/0.0
                                                 Packets                 Bytes
     Source class                      (packet-per-second)    (bits-per-second)
                              gold                 1928095            161959980
                                   (                  889) (             597762)
                            bronze                       0                    0
                                   (                    0) (                 0)
                            silver                       0                    0
                                   (                    0) (                 0)
  Logical interface so-0/1/3.0
                                                 Packets                 Bytes
     Source class                      (packet-per-second)    (bits-per-second)
                              gold                       0                    0
                                   (                    0) (                 0)
                            bronze                       0                    0
                                   (                    0) (                 0)
                            silver                  116113              9753492
                                   (                  939) (             631616)
```

show interfaces statistics

Syntax show interfaces statistics *interface-name*
<detail>

Release Information Command introduced before Junos OS Release 7.4.

Description Display static interface statistics, such as errors.

Options *interface-name*—Name of an interface.

detail—(Optional) Display detail output.

Required Privilege Level view

Related Documentation
- clear interfaces statistics on page 46

List of Sample Output

Output Fields Output from both the **show interfaces *interface-name* detail** and the **show interfaces *interface-name* extensive** commands include all the information displayed in the output from the **show interfaces statistics** command. For more information, see the particular interface type in which you are interested. For information about destination class and source class statistics, see the “Destination Class Field” section and the “Source Class Field” section under Common Output Fields Description.

Sample Output

show interfaces statistics (Fast Ethernet)

```
user@host> show interfaces fe-1/3/1 statistics
Physical interface: fe-1/3/1, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 1042
  Description: ford fe-1/3/1
  Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues     : 4 supported, 4 maximum usable queues
  Current address: 00:90:69:93:04:dc, Hardware address: 00:90:69:93:04:dc
  Last flapped   : 2006-04-18 03:08:59 PDT (00:01:24 ago)
  Statistics last cleared: Never
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms  : None
```

```
  Active defects : None
  Logical interface fe-1/3/1.0 (Index 69) (SNMP ifIndex 50)
    Flags: SNMP-Traps Encapsulation: ENET2
    Protocol inet, MTU: 1500
      Flags: Is-Primary, DCU, SCU-in
                                              Packets                 Bytes
      Destination class            (packet-per-second)    (bits-per-second)
                          silver1                    0                     0
                                  (                  0) (                  0)
                          silver2                    0                     0
                                  (                  0) (                  0)
                          silver3                    0                     0
                                  (                  0) (                  0)
      Addresses, Flags: Is-Default Is-Preferred Is-Primary
        Destination: 10.27.245/24, Local: 10.27.245.2,
        Broadcast: 10.27.245.255
    Protocol iso, MTU: 1497
      Flags: Is-Primary
```

show interfaces statistics (Gigabit Ethernet PIC—Egress)

```
user@host> show interfaces ge-5/2/0 statistics detail
Physical interface: ge-5/2/0, Enabled, Physical link is Up
  Interface index: 146, SNMP ifIndex: 519, Generation: 149
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:1d:b5:61:d9:74, Hardware address: 00:1d:b5:61:d9:74
  Last flapped   : 2009-11-11 11:24:00 PST (09:23:08 ago)
  Statistics last cleared: 2009-11-11 17:50:58 PST (02:56:10 ago)
  Traffic statistics:
   Input  bytes  :              271524                   0 bps
   Output bytes  :            37769598                 352 bps
   Input  packets:                3664                   0 pps
   Output packets:              885790                   0 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :            16681118
    Input  packets:                   0
    Output packets:              362633
   Multicast statistics:
    IPV4 multicast statistics:
     Input  bytes  :              112048                   0 bps
     Output bytes  :            20779920                   0 bps
     Input  packets:                1801                   0 pps
     Output packets:              519498                   0 pps
    IPV6 multicast statistics:
     Input  bytes  :              156500                   0 bps
     Output bytes  :            16681118                   0 bps
     Input  packets:                1818                   0 pps
     Output packets:              362633                   0 pps
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0,
    L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
```

```
 FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0,
    Resource errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort               882558               882558                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                3232                 3232                    0

  Active alarms  : None
  Active defects : None

  Logical interface ge-5/2/0.0 (Index 71) (SNMP ifIndex 573) (Generation 135)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :              271524
     Output bytes  :            37769598
     Input  packets:                3664
     Output packets:              885790
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :            16681118
      Input  packets:                   0
      Output packets:              362633
    Local statistics:
     Input  bytes  :              271524
     Output bytes  :              308560
     Input  packets:                3664
     Output packets:                3659
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :            37461038                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:              882131                    0 pps
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :            16681118
      Input  packets:                   0
      Output packets:              362633
    Multicast statistics:
     IPV4 multicast statistics:
      Input  bytes  :              112048                    0 bps
      Output bytes  :            20779920                    0 bps
      Input  packets:                1801                    0 pps
      Output packets:              519498                    0 pps
     IPV6 multicast statistics:
      Input  bytes  :              156500                    0 bps
      Output bytes  :            16681118                    0 bps
      Input  packets:                1818                    0 pps
      Output packets:              362633                    0 pps
    Protocol inet, MTU: 1500, Generation: 151, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 40.40.40.0/30, Local: 40.40.40.2, Broadcast: 40.40.40.3,
Generation: 167
    Protocol inet6, MTU: 1500, Generation: 152, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::40.40.40.0/126, Local: ::40.40.40.2
```

```
  Generation: 169
    Addresses, Flags: Is-Preferred
      Destination: fe80::/64, Local: fe80::21d:b5ff:fe61:d974
  Protocol multiservice, MTU: Unlimited, Generation: 171
  Generation: 153, Route table: 0
    Policer: Input: __default_arp_policer__
```

show interfaces statistics detail (Aggregated Ethernet)

```
user@host> show interfaces ae0 detail
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 186, SNMP ifIndex: 111, Generation: 187
  Link-level type: Ethernet, MTU: 1514, Speed: 2000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Minimum links needed: 1,
  Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:90:69:0b:2f:f0, Hardware address: 00:90:69:0b:2f:f0
  Last flapped   : Never
  Statistics last cleared: 2006-12-23 03:04:16 PST (01:16:24 ago)
  Traffic statistics:
   Input  bytes  :                28544                    0 bps
   Output bytes  :                39770                    0 bps
   Input  packets:                  508                    0 pps
   Output packets:                  509                    0 pps
   Input  bytes  :           IPv6 28544
   Output bytes  :               IPv6 0
   Input  packets:             IPv6 508
   Output packets:               IPv6 0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0,
    Policed discards: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0,
    Resource errors: 0

  Logical interface ae0.0 (Index 67) (SNMP ifIndex 139) (Generation 145)
    Flags: SNMP-Traps Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :           508          0         28544            0
        Output:           509          0         35698            0
    Link:
      ge-3/3/8.0
        Input :           508          0         28544            0
        Output:             0          0             0            0
      ge-3/3/9.0
        Input :             0          0             0            0
        Output:             0          0             0            0
    Marker Statistics:   Marker Rx     Resp Tx   Unknown Rx   Illegal Rx
      ge-3/3/8.0               0           0            0            0
      ge-3/3/9.0               0           0            0            0
  Egress queues: 8 supported, 8 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0
```

```
    Protocol inet, MTU: 1500, Generation: 166, Route table: 0
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.1.1/24, Local: 10.1.1.1, Broadcast: 10.1.1.255,
        Generation: 159
    Protocol inet6, MTU: 1500, Generation: 163, Route table: 0
      Flags: Is-Primary
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::206:5bff:fe05:c321,
        Broadcast: Unspecified, Generation: 161
```

show interfaces statistics detail (Aggregated Ethernet—Ingress)

```
user@host> show interfaces statistics detail ae0 | no-more
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 504, Generation: 278
  Link-level type: Ethernet, MTU: 1514, Speed: 1Gbps, BPDU Error: None, MAC-REWRITE
 Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Minimum links needed: 1,
Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:1d:b5:61:db:f0, Hardware address: 00:1d:b5:61:db:f0
  Last flapped   : 2009-11-09 03:30:23 PST (00:01:28 ago)
  Statistics last cleared: 2009-11-09 03:26:18 PST (00:05:33 ago)
  Traffic statistics:
   Input  bytes  :            544009602             54761856 bps
   Output bytes  :                 3396                    0 bps
   Input  packets:             11826292               148809 pps
   Output packets:                   42                    0 pps
   IPv6 transit statistics:
    Input  bytes  :           350818604
    Output bytes  :                   0
    Input  packets:             7626488
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
  Ingress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0

  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                   21                   21                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                 451                  451                    0
```

```
  Logical interface ae0.0 (Index 70) (SNMP ifIndex 574) (Generation 177)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :      11826292     148809     544009602     54761856
        Output:            42          0          3396            0
    Link:
      ge-5/2/0.0
        Input :      11826292     148809     544009602     54761856
        Output:            42          0          3396            0
    Marker Statistics:   Marker Rx     Resp Tx   Unknown Rx   Illegal Rx
      ge-5/2/0.0               0           0            0            0
    Protocol inet, MTU: 1500, Generation: 236, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 30.30.30.0/30, Local: 30.30.30.2, Broadcast: 30.30.30.3,
Generation: 310
    Protocol inet6, MTU: 1500, Generation: 237, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::30.30.30.0/126, Local: ::30.30.30.2
    Generation: 312
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::21d:b5ff:fe61:dbf0
    Protocol multiservice, MTU: Unlimited, Generation: 314
    Generation: 238, Route table: 0
      Policer: Input: __default_arp_policer__
```

show interfaces statistics detail (AggregatedEthernet—Egress)

```
user@host> show interfaces statistics detail ae0 | no-more
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 501, Generation: 319
  Link-level type: Ethernet, MTU: 1514, Speed: 1Gbps, BPDU Error: None, MAC-REWRITE
 Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Minimum links needed: 1,
Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:1f:12:c2:37:f0, Hardware address: 00:1f:12:c2:37:f0
  Last flapped   : 2009-11-09 03:30:24 PST (00:02:42 ago)
  Statistics last cleared: 2009-11-09 03:26:42 PST (00:06:24 ago)
  Traffic statistics:
   Input  bytes  :                  440                    0 bps
   Output bytes  :           1047338120             54635848 bps
   Input  packets:                    7                    0 pps
   Output packets:             22768200               148466 pps
   IPv6 transit statistics:
    Input  bytes  :                  288
    Output bytes  :            723202616
    Input  packets:                    4
    Output packets:             15721796
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
  Ingress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0
```

```
    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0

  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort            201985796            201985796                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                  65                   65                    0


  Logical interface ae0.0 (Index 72) (SNMP ifIndex 505) (Generation 204)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :             7          0           440            0
        Output:      22768200     148466    1047338120     54635848
    Link:
      ge-2/1/6.0
        Input :             7          0           440            0
        Output:      22768200     148466    1047338120     54635848
    Marker Statistics:   Marker Rx     Resp Tx   Unknown Rx   Illegal Rx
      ge-2/1/6.0               0           0            0            0
    Protocol inet, MTU: 1500, Generation: 291, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 30.30.30.0/30, Local: 30.30.30.1, Broadcast: 30.30.30.3,
Generation: 420
    Protocol inet6, MTU: 1500, Generation: 292, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::/26, Local: ::30.30.30.1
    Generation: 422
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::21f:12ff:fec2:37f0
    Protocol multiservice, MTU: Unlimited, Generation: 424
    Generation: 293, Route table: 0
      Policer: Input: __default_arp_policer__
```

show interfaces statistics (SONET/SDH)

```
user@host> show interfaces statistics detail so-3/0/0 | no-more
Physical interface: so-3/0/0, Enabled, Physical link is Up
  Interface index: 133, SNMP ifIndex: 538, Generation: 283
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC192,
 Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 13 (last seen 00:00:04 ago)
    Output: 14 (last sent 00:00:02 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Opened, iso: Not-configured, mpls: Not-configured

  CHAP state: Closed
```

```
  PAP state: Closed
  CoS queues     : 8 supported, 8 maximum usable queues
  Last flapped   : 2009-11-09 02:52:34 PST (01:12:39 ago)
  Statistics last cleared: 2009-11-09 03:58:54 PST (00:06:19 ago)
  Traffic statistics:
   Input  bytes  :           2559160294             54761720 bps
   Output bytes  :                10640                   48 bps
   Input  packets:             55633975               148809 pps
   Output packets:                  216                    0 pps
   IPv6 transit statistics:
    Input  bytes  :           647922328
    Output bytes  :                   0
    Input  packets:            14085269
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Bucket drops:
0, Policed discards: 0, L3 incompletes: 0,
    L2 channel errors: 0, L2 mismatch timeouts: 0, HS link CRC errors: 0, HS link
 FIFO overflows: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Aged packets: 0, HS link FIFO
underflows: 0, MTU errors: 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    4                    4                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                 213                  213                    0

  SONET alarms   : None
  SONET defects  : None

  Logical interface so-3/0/0.0 (Index 72) (SNMP ifIndex 578) (Generation 182)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
    Protocol inet, MTU: 4470, Generation: 244, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 30.30.30.0/30, Local: 30.30.30.2, Broadcast: 30.30.30.3,
Generation: 322
    Protocol inet6, MTU: 4470, Generation: 245, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::30.30.30.0/126, Local: ::30.30.30.2
    Generation: 324
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::2a0:a5ff:fe61:9264
    Generation: 326
```

show interfaces statistics (Aggregated SONET/SDH—Ingress)

```
user@host> show interfaces statistics detail as0 | no-more
Physical interface: as0, Enabled, Physical link is Up
  Interface index: 132, SNMP ifIndex: 534, Generation: 282
  Link-level type: PPP, MTU: 4474, Speed: OC192, Minimum links needed: 1, Minimum
 bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Last flapped   : 2009-11-09 03:45:53 PST (00:09:38 ago)
  Statistics last cleared: 2009-11-09 03:48:17 PST (00:07:14 ago)
```

```
  Traffic statistics:
   Input  bytes  :          2969786332             54761688 bps
   Output bytes  :               11601                    0 bps
   Input  packets:            64560636               148808 pps
   Output packets:                 225                    0 pps
   IPv6 transit statistics:
    Input  bytes  :          2086013152
    Output bytes  :                   0
    Input  packets:            45348114
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    3                    3                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                 222                  222                    0


  Logical interface as0.0 (Index 71) (SNMP ifIndex 576) (Generation 179)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :      64560550     148808    2969785300     54761688
        Output:           139          0         10344            0
    Link:
      so-3/0/0.0
        Input :      64560550     148808    2969785300     54761688
        Output:           139          0         10344            0
    Protocol inet, MTU: 4470, Generation: 240, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 30.30.30.0/30, Local: 30.30.30.2, Broadcast: 30.30.30.3,
Generation: 316
    Protocol inet6, MTU: 4470, Generation: 241, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::30.30.30.0/126, Local: ::30.30.30.2
    Generation: 318
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::2a0:a5ff:fe61:9264
    Generation: 320
```

show interfaces statistics (Aggregated SONET/SDH—Egress)

```
user@host> show interfaces statistics detail as0 | no-more
Physical interface: as0, Enabled, Physical link is Up
  Interface index: 132, SNMP ifIndex: 565, Generation: 323
  Link-level type: PPP, MTU: 4474, Speed: OC192, Minimum links needed: 1, Minimum
 bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Last flapped   : 2009-11-09 03:43:37 PST (00:12:48 ago)
  Statistics last cleared: 2009-11-09 03:48:54 PST (00:07:31 ago)
```

```
  Traffic statistics:
   Input  bytes  :                11198                 392 bps
   Output bytes  :           3101452132            54783448 bps
   Input  packets:                  234                   0 pps
   Output packets:             67422937              148868 pps
   IPv6 transit statistics:
    Input  bytes  :                5780
    Output bytes  :          2171015678
    Input  packets:                  72
    Output packets:            47195993
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort             67422830             67422830                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                  90                   90                    0


  Logical interface as0.0 (Index 71) (SNMP ifIndex 548) (Generation 206)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :           144          0         10118          392
        Output:      67422847     148868    3101450962     54783448
    Link:
      so-0/1/0.0
        Input :           144          0         10118          392
        Output:      67422847     148868    3101450962     54783448
    Protocol inet, MTU: 4470, Generation: 295, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 30.30.30.0/30, Local: 30.30.30.1, Broadcast: 30.30.30.3,
Generation: 426
    Protocol inet6, MTU: 4470, Generation: 296, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: ::/26, Local: ::30.30.30.1
    Generation: 428
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::2a0:a5ff:fe63:1d0a
    Generation: 429
```

show route

Syntax

show route
<all>
<*destination-prefix*>
<logical-system (all | *logical-system-name*)>
<private>

Syntax (EX Series Switch)

show route
<all>
<*destination-prefix*>
<private>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
private option introduced in Junos OS Release 9.5.
private option introduced in Junos OS Release 9.5 for EX Series switches.

Description

Display the active entries in the routing tables.

Options

none—Display brief information about all active entries in the routing tables.

all—(Optional) Display information about all routing tables, including private, or internal, routing tables.

destination-prefix—(Optional) Display active entries for the specified address or range of addresses.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

private—(Optional) Display information only about all private, or internal, routing tables.

Required Privilege Level

view

List of Sample Output

show route on page 66
show route destination-prefix on page 66
show route extensive on page 66

Output Fields

Table 3 on page 63 describes the output fields for the **show route** command. Output fields are listed in the approximate order in which they appear.

Table 3: show route Output Fields

Field Name	Field Description
routing-table-name	Name of the routing table (for example, **inet.0**).
number **destinations**	Number of destinations for which there are routes in the routing table.

Table 3: show route Output Fields *(continued)*

Field Name	Field Description
***number* routes**	Number of routes in the routing table and total number of routes in the following states: • **active** (routes that are active). • **holddown** (routes that are in the pending state before being declared inactive). • **hidden** (routes that are not used because of a routing policy).

destination-prefix	Route destination (for example:**10.0.0.1/24**). Sometimes the route information is presented in another format, such as:
	• ***MPLS-label*** (for example, **80001**).
	• ***interface-name*** (for example, **ge-1/0/2**).
	• ***neighbor-address*:*control-word-status*:*encapsulation type*:*vc-id* :*source*** (Layer 2 circuit only; for example, **10.1.1.195:NoCtrlWord:1:1:Local/96**):
	  ◦ ***neighbor-address***—Address of the neighbor.
	  ◦ ***control-word-status***—Whether the use of the control word has been negotiated for this virtual circuit: **NoCtrlWord** or **CtrlWord**.
	  ◦ ***encapsulation type***—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport.
	  ◦ ***vc-id***—Virtual circuit identifier.
	• ***source***—Source of the advertisement: **Local** or **Remote**.

[*protocol*, *preference*]	Protocol from which the route was learned and the preference value for the route. • **+**—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • **-** —A hyphen indicates the last active route. • *****—An asterisk indicates that the route is both the active and the last active route. An asterisk before a **to** line indicates the best subpath to the route. In every routing metric except for the BGP **LocalPref** attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the **LocalPref** value in the **Preference2** field. For example, if the **LocalPref** value for Route 1 is 100, the **Preference2** value is -101. If the **LocalPref** value for Route 2 is 155, the **Preference2** value is -156. Route 2 is preferred because it has a higher **LocalPref** value and a lower **Preference2** value.
weeks:days hours:minutes:seconds	How long the route been known (for example, **2w4d 13:11:14**, or 2 weeks, 4 days, 13 hours, 11 minutes and 14 seconds).
metric	Cost value of the indicated route. For routes within an AS, the cost is determined by IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.
localpref	Local preference value included in the route.
from	Interface from which the route was received.

Table 3: show route Output Fields *(continued)*

Field Name	Field Description
AS path	AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated: • **I**—IGP. • **E**—EGP. • **?**—Incomplete; typically, the AS path was aggregated. When AS path numbers are included in the route, the format is as follows: • **[ ]**—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device, or if AS path prepending is configured. • **{ }**—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • **( )**—Parentheses enclose a confederation. • **([ ])**—Parentheses and brackets enclose a confederation set. NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.
to	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.
via	Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word **Selected**. This field can also contain the following information: • **Weight**—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when Multiprotocol Label Switching (MPLS) label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • **Balance**—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable Border Gateway Protocol (BGP) multipath load balancing. • ***lsp-path-name***—Name of the label-switched path (LSP) used to reach the next hop. • ***label-action***—MPLS label and operation occurring at the next hop. The operation can be **pop** (where a label is removed from the top of the stack), **push** (where another label is added to the label stack), or **swap** (where a label is replaced by another label).

Sample Output

show route

```
user@host> show route
inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 1w5d 20:30:29
                      Discard
10.255.245.51/32   *[Direct/0] 2w4d 13:11:14
                    > via lo0.0
172.16.0.0/12      *[Static/5] 2w4d 13:11:14
                    > to 192.168.167.254 via fxp0.0
192.168.0.0/18     *[Static/5] 1w5d 20:30:29
                    > to 192.168.167.254 via fxp0.0
192.168.40.0/22    *[Static/5] 2w4d 13:11:14
                    > to 192.168.167.254 via fxp0.0
192.168.64.0/18    *[Static/5] 2w4d 13:11:14
                    > to 192.168.167.254 via fxp0.0
192.168.164.0/22   *[Direct/0] 2w4d 13:11:14
                    > via fxp0.0
192.168.164.51/32  *[Local/0] 2w4d 13:11:14
                      Local via fxp0.0
207.17.136.192/32  *[Static/5] 2w4d 13:11:14
                    > to 192.168.167.254 via fxp0.0

green.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100.101.0.0/16     *[Direct/0] 1w5d 20:30:28
                    > via fe-0/0/3.0
100.101.2.3/32     *[Local/0] 1w5d 20:30:28
                      Local via fe-0/0/3.0
224.0.0.5/32       *[OSPF/10] 1w5d 20:30:29, metric 1
                      MultiRecv

red.inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.10.10.10/32     *[Direct/0] 01:08:46
                    > via lo0.1
10.255.245.212/32  *[BGP/170] 00:01:40, localpref 100, from 10.255.245.204
                      AS path: 300 I
                    > to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix
10.255.245.213/32  *[BGP/170] 00:40:47, localpref 100
                      AS path: 100 I
                    > to 100.1.1.1 via so-0/0/1.0
```

show route destination-prefix

```
user@host> show route 172.16.0.0/12

inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.0/12      *[Static/5] 2w4d 12:54:27
                    > to 192.168.167.254 via fxp0.0
```

show route extensive

```
user@host> show route extensive
inet.0: 335844 destinations, 335845 routes (335395 active, 0 holddown, 450 hidden)
1.9.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 1.9.0.0/16 -> {indirect(342)}
Page 0 idx 1 Type 1 val db31a80
    Nexthop: Self
    AS path: [69] 10458 14203 2914 4788 4788 I
```

```
    Communities: 2914:410 2914:2403 2914:3400
Path 1.9.0.0 from 192.168.69.71 Vector len 4.  Val: 1
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 1006553
                Source: 192.168.69.71
                Next hop type: Router, Next hop index: 324
                Next hop: 192.168.167.254 via fxp0.0, selected
                Protocol next hop: 192.168.69.71
                Indirect next hop: 8e166c0 342
                State: <Active Ext>
                Local AS:    69 Peer AS: 10458
                Age: 6d 10:58:10        Metric2: 0
                Task: BGP_10458.192.168.69.71+179
                Announcement bits (3): 0-KRT 2-BGP RT Background 3-Resolve tree
1
                AS path: 10458 14203 2914 4788 4788 I
                Communities: 2914:410 2914:2403 2914:3400
                Accepted
                Localpref: 100
                Router ID: 207.17.136.192
                Indirect next hops: 1
                        Protocol next hop: 192.168.69.71
                        Indirect next hop: 8e166c0 342
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 192.168.167.254 via fxp0.0
                        192.168.0.0/16 Originating RIB: inet.0
                          Node path count: 1
                          Forwarding nexthops: 1
                                Nexthop: 192.168.167.254 via fxp0.0
```

PART 5

Index

Index