



Junos OS

Junos Telemetry Interface Feature Guide



Modified: 2017-12-07

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos OS Junos Telemetry Interface Feature Guide

Copyright © 2017 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	ix
	Documentation and Release Notes	ix
	Supported Platforms	ix
	Using the Examples in This Manual	ix
	Merging a Full Example	x
	Merging a Snippet	x
	Documentation Conventions	xi
	Documentation Feedback	xiii
	Requesting Technical Support	xiii
	Self-Help Online Tools and Resources	xiii
	Opening a Case with JTAC	xiv
Part 1	Junos Telemetry Interface Overview	
Chapter 1	Overview of the Junos Telemetry Interface	3
	Overview of the Junos Telemetry Interface	4
	Telemetry Sensors and Data Models	4
	Uses and Benefits	5
Part 2	Configuring Native Sensors	
Chapter 2	Export Format of Collected Data	9
	Understanding the Junos Telemetry Interface Export Format of Collected	
	Data	10
	Understanding the Sensor Data Encapsulation Format	10
Chapter 3	Configuring Junos Telemetry Interface (CLI Procedure)	15
	Configuring a Junos Telemetry Interface Sensor (CLI Procedure)	15
	Configuring an Export Profile	16
	Configuring a Streaming Server Profile	19
	Configuring a Sensor Profile	20
	Verifying Junos Telemetry Interface Sensor Configuration	21
Chapter 4	Junos Telemetry Interface Configuration Statements and Operational	
	Commands	25
	export-profile (Junos Telemetry Interface)	26
	per-interface-per-member-link	30
	per-sid	31
	sensor (Junos Telemetry Interface)	32
	sensor-based-stats (Junos Telemetry Interface)	39
	streaming-server (Junos Telemetry Interface)	40
	show agent sensors	42

Chapter 5	Decoding Data	45
	Decoding Junos Telemetry Interface Data With UNIX Utilities	45
	Preparing the Collector to Decode Data	45
	Decoding Data on the Collector	46
Part 3	Configuring gRPC Sensors	
Chapter 6	OpenConfig and gRPC for Junos Telemetry Interface	57
	Understanding OpenConfig and gRPC on Junos Telemetry Interface	58
	Network Agent Software	58
	Using OpenConfig for Junos OS to Enable Junos Telemetry Interface	58
	Using gRPC to Stream Data	59
	Installing the Network Agent Package (Junos Telemetry Interface)	61
	gRPC Services for Junos Telemetry Interface	64
	Configuring gRPC for the Junos Telemetry Interface	64
	Configuring Bidirectional Authentication for gRPC for Junos Telemetry Interface	66
	ssl	69
	Guidelines for gRPC Sensors (Junos Telemetry Interface)	70
	Supported gRPC Sensors	71
	Understanding YANG on Devices Running Junos OS	143
	Configurable NETCONF Proxy for Junos Telemetry Interface	144
	Creating a User-Defined YANG File	144
	Example: Kernel Routing Table (KRT) Statistics	146
	Installing a User-Defined YANG File	149
	request system yang add	151
	request system yang delete	153
	request system yang update	155
	request system yang validate	157
Part 4	Best Practices	
Chapter 7	Best Practices for Implementing the Junos Telemetry Interface	161
	Guidelines for Specifying Data Reporting Intervals Junos Telemetry Interface . . .	161
	How to Determine the Reporting Interval for a System Resource	161
	Guidelines for Aggregating Junos Telemetry Interface Data	162
	Aggregating Data Over Fixed Time Spans	162
	Example: Aggregating Data for Gauge Metrics	162
	Example: Aggregating Data for Cumulative Statistics	163
	Aggregating Data From Multiple Sources	164
	Example: Aggregating Data from Multiple Sources	165
	Aggregating Data for Multiple Metrics	165
	Example: Aggregating Multiple Metric Values	166

List of Figures

Part 1	Junos Telemetry Interface Overview	
Chapter 1	Overview of the Junos Telemetry Interface	3
	Figure 1: Telemetry Streaming for Performance Management	5

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xii
Part 2	Configuring Native Sensors	
Chapter 2	Export Format of Collected Data	9
	Table 3: Individual Data Element Types in the gpb Message	13
Chapter 4	Junos Telemetry Interface Configuration Statements and Operational Commands	25
	Table 4: resource statement Options	34
	Table 5: show agent sensors Output Fields	42
Part 3	Configuring gRPC Sensors	
Chapter 6	OpenConfig and gRPC for Junos Telemetry Interface	57
	Table 6: Telemetry RPCs	59
	Table 7: gRPC Sensors	71
	Table 8: Broadband Edge gRPC Sensors	105
Part 4	Best Practices	
Chapter 7	Best Practices for Implementing the Junos Telemetry Interface	161
	Table 9: Telemetry Data Values	163

About the Documentation

- [Documentation and Release Notes on page ix](#)
- [Supported Platforms on page ix](#)
- [Using the Examples in This Manual on page ix](#)
- [Documentation Conventions on page xi](#)
- [Documentation Feedback on page xiii](#)
- [Requesting Technical Support on page xiii](#)

Documentation and Release Notes

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

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

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

Supported Platforms

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

- [MX Series](#)
- [PTX Series](#)
- [QFX Series](#)
- [EX Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

```
commit {
```

```
file ex-script-snippet.xml; }
```

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

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

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







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

For more information about the **load** command, see [CLI Explorer](#).

Documentation Conventions

[Table 1 on page xi](#) defines notice icons used in this guide.

Table 1: Notice Icons

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

[Table 2 on page xii](#) defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

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

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Junos Telemetry Interface Overview

- Overview of the Junos Telemetry Interface on page 3

CHAPTER 1

Overview of the Junos Telemetry Interface

- Overview of the Junos Telemetry Interface on page 4

Overview of the Junos Telemetry Interface

As the number of objects on the network and the metrics they generate have grown, the traditional models, such as SNMP, used to gather operational statistics for monitoring the health of a network, have imposed limits on network element scale and efficiency. The so-called pull model used by SNMP and the CLI, which requires additional processing to periodically poll the network element, directly limits scaling.

The Junos Telemetry Interface (JTI) overcomes these limits by relying on a so-called push model to deliver data asynchronously, which eliminates polling. A request to send data is sent once by a management station to stream periodic updates. As a result, JTI is highly scalable and can support the monitoring of thousands of objects in a network.



NOTE: Junos Telemetry Interface was introduced in Junos OS Release 15.1F3, on MX Series routers with interfaces configured on MPC1 through MPC6E, and on PTX Series routers with interfaces configured on FPC3. Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1, FPC2, and dual Routing Engines on PTX Series routers are also supported.

Starting with Junos OS Release 17.2R1, QFX10000 and QFX5200 switches, and PTX1000 routers are also supported. QFX5200 switches support only gRPC sensors.

Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600 and EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers are also supported. QFX5110 switches support only gRPC sensors.

Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are supported.

-
- [Telemetry Sensors and Data Models on page 4](#)
 - [Uses and Benefits on page 5](#)

Telemetry Sensors and Data Models

The Junos Telemetry Interface enables you to provision sensors to collect and export data for various system resources, such as physical interfaces and firewall filters. Two data models, each of which uses a different mode of transport, are supported:

- An open and extensible data model defined by Juniper Networks. Data is generated as Google protocol buffers (gpb) structured messages. The files that define each **.proto** message are published on the Juniper Networks web site. Native sensors export data close to the source, such as the line card or network processing unit (NPU), using the User Datagram Protocol (UDP). Because this model features a distributed architecture, it scales easily.

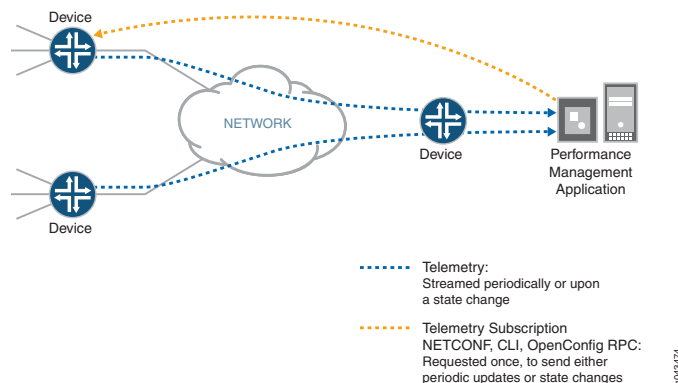
- An OpenConfig data model that generates data as gpb messages in a universal key/value format. OpenConfig for Junos OS, which you must download, supports the YANG data models. gRPC remote procedure calls (gRPC) are used to provision sensors and to subscribe to and receive telemetry data. gRPC is based on TCP, and supports SSL encryption, so it is considered secure and reliable. If your Juniper Networks device is running a version of Junos OS with the upgraded FreeBSD kernel, this model requires you to download the Junos Network Agent package, which runs on the Routing Engine and provides interfaces to manage gRPC subscriptions. For other versions of Junos OS, Network Agent functionality is embedded in the software.

Uses and Benefits

A primary function of the Junos Telemetry Interface is performance monitoring. Streaming data to a performance management system enables network administrators to measure trends in link and node utilization, and troubleshoot such issues as network congestion in real time, for example.

In a typical deployment, the network element, or device, streams duplicate data to two destination servers that function as performance management system collectors. Streaming data to two collectors provides redundancy. See [Figure 1 on page 5](#) for an illustration of how the performance management system collectors request data and how the device streams data. The device provisions sensors to collect and export data using command-line interface (CLI), configuration through NETCONF, or gRPC subscription calls. The collectors request data by initiating a telemetry subscription. Data is requested only once and is streamed periodically.

Figure 1: Telemetry Streaming for Performance Management



Other applications of the Junos Telemetry Interface include providing real-time data to support operational state synchronization between a network element and an external controller, such as the Northstar Controller, which automates the creation of traffic-engineering paths across the network. The NorthStar Controller can subscribe to telemetry data about certain network elements, such as label-switched path (LSP) statistics.

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are supported.
17.3R1	Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600 and EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers are also supported. QFX5110 switches support only gRPC sensors.
17.2R1	Starting with Junos OS Release 17.2R1, QFX10000 and QFX5200 switches, and PTX1000 routers are also supported. QFX5200 switches support only gRPC sensors.
16.1R3	Starting with Junos OS Release 16.1R3, FPC1, FPC2, and dual Routing Engines on PTX Series routers are also supported.
15.1F5	Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.
15.1F3	Junos Telemetry Interface was introduced in Junos OS Release 15.1F3, on MX Series routers with interfaces configured on MPC1 through MPC6E, and on PTX Series routers with interfaces configured on FPC3.

Related Documentation

- [Understanding the Junos Telemetry Interface Export Format of Collected Data on page 10](#)
- [Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 58](#)

PART 2

Configuring Native Sensors

- [Export Format of Collected Data on page 9](#)
- [Configuring Junos Telemetry Interface \(CLI Procedure\) on page 15](#)
- [Junos Telemetry Interface Configuration Statements and Operational Commands on page 25](#)
- [Decoding Data on page 45](#)

CHAPTER 2

Export Format of Collected Data

- Understanding the Junos Telemetry Interface Export Format of Collected Data on page 10

Understanding the Junos Telemetry Interface Export Format of Collected Data

The Junos Telemetry Interface supports two ways of exporting data in the Google protocol buffers (gpb) format:

- Through UDP from so-called native sensors that export data close to the source, such as the line card or network processing unit (NPU). Juniper Networks defines the data model, which is open and extensible.
- Through gRPC remote procedure calls (gRPC) that export data through the Routing Engine. The data model is defined by OpenConfig, which supports the use of vendor-neutral data models to configure and manage the network. OpenConfig for Junos OS supports the YANG data models. For platforms that are running a version of Junos OS based on an upgraded FreeBSD kernel only, you must install a separate package called Network Agent that functions as a gRPC server and terminates the RPC interfaces. For all other versions of Junos OS, the Network Agent functionality is embedded in the software. You must also install the OpenConfig for Junos OS module and the YANG models.

This section describes the format of data exported from native sensors using UDP. The data is encapsulated into a UDP header, which is in turn encapsulated in the IPv4 payload. This model of the Junos Telemetry Interface is based on a distributed architecture, through which the data generated by configured sensors is exported directly from the data plane, bypassing the control plane, and thus conserving these resources to perform other necessary functions.



NOTE: The Junos Telemetry Interface was introduced in Junos OS Release 15.1F3, on MX Series routers with interfaces configured on MPC1 through MPC6E, and on PTX Series routers with interfaces configured on FPC3. Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1, FPC2, and dual Routing Engines on PTX Series routers are also supported.

Starting with Junos OS Release 17.2R1, QFX10000 and QFX5200 switches are also supported. On QFX5200 switches, only gRPC streaming is supported.

Starting with Junos OS Release 17.3R1, QFX5110 switches, EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers are also supported. On QFX5110 switches, only gRPC streaming is supported.

-
- [Understanding the Sensor Data Encapsulation Format on page 10](#)

Understanding the Sensor Data Encapsulation Format

A native sensor exports data close to the source using UDP. Various types of telemetry data, such as physical interface statistics, firewall filter counter statistics, or statistics

for label-switched paths (LSPs) can be exported. A sensor starts to emit data as soon as it is enabled.

The sensor data is represented as a single structured Google protocol buffers message, named **TelemetryStream**. The message, or **.proto** file, shown below, includes several attributes that identify the data source, such as a line card, a Packet Forwarding Engine, or a Routing Engine. The name of the configured sensor is also included. For more information about how to configure sensors, see [“Configuring a Junos Telemetry Interface Sensor \(CLI Procedure\)” on page 15](#). For a list of supported native sensors, see [sensor](#).

You must also download the **.proto** files for all the sensors supported to a streaming server or collector. From a Web browser, navigate to the All Junos Platforms software download URL on the Juniper Networks page: <http://www.juniper.net/support/downloads/>. After you select the name of the Junos OS platform and the release number, go to the **Tools** section and download the **Junos Telemetry Interface Data Model Files** package. For more information about configuring a streaming-server, see [streaming-server \(Junos Telemetry Interface\)](#).

Google protocol buffers message Definition

Following is the message definition for **TelemetryStream** in the Google Protocol Buffers definition language. It shows several optional nested structures, such as **EnterpriseSensors**, which carry privately defined sensor data.

```
//
// This file defines the top level message used for all Juniper
// Telemetry packets encoded to the protocol buffer format.
// The top level message is TelemetryStream.
//

import "google/protobuf/descriptor.proto";

extend google.protobuf.FieldOptions {
    optional TelemetryFieldOptions telemetry_options = 1024;
}

message TelemetryFieldOptions {
    optional bool is_key           = 1;
    optional bool is_timestamp     = 2;
    optional bool is_counter       = 3;
    optional bool is_gauge         = 4;
}

message TelemetryStream {
    // router name or export IP address
    required string system_id      = 1 [(telemetry_options).is_key = true];

    // line card / RE (slot number)
    optional uint32 component_id   = 2 [(telemetry_options).is_key = true];

    // PFE (if applicable)
    optional uint32 sub_component_id = 3 [(telemetry_options).is_key = true];

    // configured sensor name
    optional string sensor_name    = 4 [(telemetry_options).is_key = true];

    // sequence number, monotonically increasing for each
```

```

// system_id, component_id, sub_component_id + sensor_name.
optional uint32 sequence_number = 5;

// timestamp (milliseconds since 00:00:00 UTC 1/1/1970)
optional uint64 timestamp = 6 [(telemetry_options).is_timestamp =
true];

// major version
optional uint32 version_major = 7;

// minor version
optional uint32 version_minor = 8;

optional IETFSensors ietf = 100;

optional EnterpriseSensors enterprise = 101;
}

message IETFSensors {
    extensions 1 to max;
}

message EnterpriseSensors {
    extensions 1 to max;
}

extend EnterpriseSensors {
    // re-use IANA assigned numbers
    optional JuniperNetworksSensors juniperNetworks = 2636;
}

message JuniperNetworksSensors {
    extensions 1 to max;
}

```

The **TelemetryStream** message also includes optional nested structures that carry different types of data. One structure carries enterprise, that is, privately defined data. Individual companies, such as Juniper Networks, define and maintain the attributes generated by enterprise sensors. Each company is assigned a unique attribute identifier. The current convention is to use IANA-assigned enterprise MIB identifiers for each attribute. For Juniper Networks, this assigned identifier is 2636.



BEST PRACTICE: To verify that a particular message type has been exported and received, check for those attributes under **TelemetryStream.enterprise.juniperNetworks** in the gpb message.

See [Table 3 on page 13](#) for descriptions of each element collected by sensor data, including semantics and corresponding schema.

Table 3: Individual Data Element Types in the gpb Message

Element Type	Description
Counter	An unsigned integer that increases monotonically. When it reaches its maximum value, it starts back at zero.
Gauge	An unsigned 32-bit or 64-bit integer that can increase or decrease in value. An example of the data represented by this element is the instantaneous value of a specific resource, such as queue depth or temperature.
Rate	Rate at which a base metric changes, such as a counter or a gauge. For this element type, units of measurement are defined explicitly (such as bits per second), as well as the interval over which the rate is collected.
Average	The average of several samples of a base metric. For example, an <i>average queue depth</i> data element would be calculated by averaging several elements of the queue depth. For this element type, we strongly recommend defining the number of measurements used to compute the average, as well as the time interval between the measurements. Otherwise, you should define explicitly the means by which this average value is calculated.
Peak	Maximum value among several samples of a base metric. For example, a <i>peak queue depth</i> element would be calculated by comparing several measurements of the queue depth and selecting the maximum. For this data element type, we strongly recommend that you define the number of measurements used to compute the peak value, as well as the time interval between measurements. Otherwise, define explicitly how this peak value is defined. You must also know whether this value is never cleared and thus represents the overall maximum value over all time.



NOTE: Each data element type also includes element subsets. For example, the data elements Counter and Gauge would include subsets for rate, average, and peak measurements.

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, QFX5110 switches, EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers also supported. On QFX5110 switches, only gRPC streaming is supported.
17.2R1	Starting with Junos OS Release 17.2R1, QFX10000 and QFX5200 switches are also supported. On QFX5200 switches, only gRPC streaming is supported.
16.1R3	Starting with Junos OS Release 16.1R3, FPC1, FPC2, and dual Routing Engines on PTX Series routers are also supported.
15.1F5	Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.
15.1F3	The Junos Telemetry Interface was introduced in Junos OS Release 15.1F3, on MX Series routers with interfaces configured on MPC1 through MPC6E, and on PTX Series routers with interfaces configured on FPC3.

Related Documentation

- [Decoding Junos Telemetry Interface Data With UNIX Utilities on page 45](#)

CHAPTER 3

Configuring Junos Telemetry Interface (CLI Procedure)

- [Configuring a Junos Telemetry Interface Sensor \(CLI Procedure\) on page 15](#)

Configuring a Junos Telemetry Interface Sensor (CLI Procedure)

Junos Telemetry Interface provides for the highly scalable streaming of telemetry information. Unlike previous monitoring systems, such as SNMP, which use the so-called pull model, the Junos Telemetry Interface uses the push model to collect data. The push model overcomes earlier scaling limits and reduces the processing required by the management station. You can enable monitoring and streaming of data for various system resources, such as physical and logical interfaces and firewall filters. To monitor a specific system resource, you configure a sensor. Each sensor configuration requires three main components:

- Sensor profile—Enables the system resource to monitor and allows you to set related parameters, such as the destination server to send data.
- Export profile—Specifies the attributes for the process of exporting collected data, such as the transport protocol to use and the interval at which to collect data.
- Streaming server profile—Specifies the server for collecting data and related parameters, including the destination IP address and port number.



NOTE: Junos Telemetry Interface was introduced in Junos OS Release 15.1F3 on MX Series routers with interfaces configured on MPC1 through MPC6E and on PTX Series routers with interfaces configured on FPC3. Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1 and FPC2 on PTX Series routers are also supported.

Starting with Junos OS Release 17.2R1, QFX10000 and PTX1000 switches are also supported.

Starting with Junos OS Release 17.3R1, EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers are also supported.

Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are supported. All sensors are supported except for those for fabric statistics and high queue-scale statistics.



BEST PRACTICE: We recommend that you configure at least one export profile and at least one streaming server before you configure a sensor profile. This way you can associate an export profile and a streaming server with the sensor profile configuration.

Before you begin:

- Configure a connection from your Juniper Networks device to a server that is using in-band management interfaces.
- [Configuring an Export Profile on page 16](#)
- [Configuring a Streaming Server Profile on page 19](#)
- [Configuring a Sensor Profile on page 20](#)
- [Verifying Junos Telemetry Interface Sensor Configuration on page 21](#)

Configuring an Export Profile

An export profile defines the parameters of the export process of data generated through the Junos Telemetry Interface. You must configure at least one export profile, but you can configure multiple export profiles. Each export profile can be associated with multiple sensor profiles. However, you can associate only one export profile with a specific sensor profile.



NOTE: Starting with Junos OS Release 17.3R1 on MX Series routers only, you can specify a packet loss priority for an export profile. As a result, you can apply the appropriate packet loss priority to each sensor. Loss priority settings help determine which packets are dropped from the network during periods of congestion. Previously, you could specify only the forwarding class and the DSCP value in an export profile. The following packet loss priority settings are supported: high, low, medium-high and medium-low. For more information about packet loss priority settings, see *Mapping PLP to RED Drop Profiles*.

To configure an export profile:

1. Specify a name for the export profile.

```
[edit services analytics]
user@host# set export-profile name
```

For example, to specify an export-profile name of **export-params**:

```
[edit services analytics]
user@host# set export-profile export-params
```

2. Specify the source IP address of exported packets.

```
[edit services analytics export-profile name]
user@host# set local-address ip-address
```

For example, to specify a source IP address of 192.0.2.3 for an export profile with the name **export-params**:

```
[edit services analytics export-profile export-params]
user@host# set local-address 192.0.2.3
```

3. Specify the source port number of exported packets.

```
[edit services analytics export-profile name]
user@host# set local-port number
```

For example, to specify a source port number of 21111 for an export profile with the name **export-params**:

```
[edit services analytics export-profile export-params]
user@host# set local-port 21111
```

4. Specify the interval, in seconds, at which the sensor generates telemetry data.

```
[edit services analytics export-profile name]
user@host# set reporting-rate seconds
```

For example, to specify an interval of 20 seconds at which any sensor associated with the export-profile with the name **export-params** generates telemetry data :

```
[edit services analytics sensor export-profile export-params]
user@host# set reporting-rate 20
```

5. Specify the format to define the structure of the exported data.



NOTE: The only currently supported format is Google protocol buffers (gpb)

```
[edit services analytics export-profile name]  
user@host# set format gpb
```

For example, to specify the Google protocol buffers format for exported data for an export-profile with the name **export-params**:

```
[edit services analytics export-profile export-params]  
user@host# set format gpb
```

6. Specify the transport protocol to carry the telemetry data in the IP packets.

```
[edit services analytics export-profile name]  
user@host# set transport protocol-name
```

For example, to specify the UDP as the transport protocol for telemetry data for an export profile with the name **export-params**:

```
[edit services analytics export-profile export-params]  
user@host# set transport udp
```

7. (Optional) Specify the DiffServ code point (DSCP) value to assign to exported packets.



NOTE: The default value is 0 (zero).

Any interface-level DSCP rewrite rules you have configured override the DSCP value you specify for the export profile. You need to specify a DSCP value for the export profile only if you do not configure DSCP rewrite rules on the outgoing interface. For more information, see *Configuring Rewrite Rules*.

```
[edit services analytics export-profile name]  
user@host# set dscp value
```

For example, to specify a DSCP value of 20 for an export profile with the name **export-params**:

```
[edit services analytics export-profile export-params]  
user@host# set dscp 20
```

8. (Optional) Specify a forwarding class to assign to exported packets.



NOTE: You can specify a forwarding class only for packets exported by Packet Forwarding Engine sensors. The default value is **best-effort**.

```
[edit services analytics export-profile name]
```



```
user@host# set forwarding-class class-name
```

For example, to specify a forwarding class of **assured-forwarding** for an export-profile with the name **export-params**:

```
[edit services analytics export-profile export-params]
user@host# set forwarding-class assured forwarding
```

9. (Optional) (MX Series routers only on Junos OS Release 17.3R1 or later) Specify a packet loss priority to assign to exported packets.

```
[edit services analytics export-profile name]
user@host# set loss-priority (low | high | medium-low | medium-high)
```

For example, to specify a loss priority of **high** for an export profile with the name **export-params**:

```
[edit services analytics export-profile export-params]
user@host# set loss-priority high
```

Configuring a Streaming Server Profile

A server profile defines the parameters of the server that collects exported telemetry data. You can define more than one server profile. You can also associate the same server profile with more than one sensor profile. Starting in Junos OS Release 15.1F6, you can associate more than one server with a specific sensor.

To define the profile of a streaming server to collect exported telemetry data:

1. Specify the name of the streaming sever.

```
[edit services analytics]
user@host# set streaming-server server-name
```

For example, to specify a streaming-server name of **telemetry server**:

```
[edit services analytics]
user@host# set streaming-server telemetry-server
```

2. Specify a destination IP address for the exported packets.

```
[edit services analytics streaming-server server-name]
user@host# set remote-address ip-address
```

For example, to specify a destination address of 192.0.2.2 for a streaming server with the name **telemetry-server**:

```
[edit services analytics streaming-server telemetry-server]
user@host# set remote-address 192.0.2.2
```

3. Specify a destination port number for the exported packets.

```
[edit services analytics streaming-server server-name]
user@host# set remote-port number
```

For example, to specify a destination port number of 30000 for a streaming server with the name **telemetry-server**:

```
[edit services analytics streaming-server telemetry-server]
user@host# set remote-port 30000
```

Configuring a Sensor Profile

A sensor profile defines the parameters of the system resource to monitor and stream data. You can enable only one system resource to monitor for each sensor profile. Configure a different sensor profile for each system resource you want to monitor. You can, however, configure more than one sensor to monitor the same system resource. For example, you might want to configure different parameters for exporting data for the same system resource.

To configure a sensor profile:

1. Specify the name of the sensor.

```
[edit services analytics]
user@host# set sensor sensor-name
```

For example, to specify a sensor name of **interface-1**:

```
[edit services analytics]
user@host# set sensor interface-1
```

2. Specify the system resource to monitor and stream data.

```
[edit services analytics sensor sensor-name]
user@host# set resource resource-string-identifier
```

For example, to enable monitoring of logical interfaces for sensor **interface-1**:

```
[edit services analytics sensor interface-1]
user@host# set resource /junos/system/linecard/interface/logical/usage/
```



NOTE: You must enter the resource string exactly.

3. (Optional) Specify a regular expression to filter data for the system resource you specified in Step 2. If you do not specify a regular expression, the system resource is monitored globally, that is, systemwide.

```
[edit services analytics sensor sensor-name]
user@host# set resource-filter regular-expression
```

For example, to filter data only for Ethernet logical interfaces for sensor **interface-1**:

```
[edit services analytics sensor interface-1]
user@host# set resource-filter et-*
```

4. Specify the name of a export profile configured at the **[edit export-profile *profile-name*]** hierarchy level to associate with the sensor profile. This export profile defines the parameters for exporting telemetry data.

```
[edit services analytics sensor sensor-name]
user@host# set export-name export-profile-name
```

For example, to associate an export profile named **export-params** with a sensor named **interface-1**:

```
[edit services analytics sensor interface-1]
user@host# set export-name export-params
```

5. Specify the name of a streaming server name configured at the **[edit services analytics streaming-server *server-name*]** hierarchy level to collect exported data.



NOTE: Starting in Junos OS Release 15.1F6, you can specify more than one streaming server for a sensor profile. To specify more than one streaming server for a sensor, you must enclose the names in brackets.

```
[edit services analytics sensor sensor-name]
user@host# set streaming-server server-name
```

For example, to associate a streaming server name **telemetry-server** with a sensor named **interface-1**:

```
[edit services analytics sensor interface-1]
user@host# set streaming-server telemetry-server
```

Verifying Junos Telemetry Interface Sensor Configuration

Purpose Confirm your configuration.

Action From configuration mode, confirm your configuration by entering the **show services analytics** command. If your output does not display the intended configuration, repeat the instructions in this configuration procedure to correct the configuration.

```
user@host# show services analytics
streaming-server telemetry-server {
  remote-address 192.0.2.2;
  remote-port 30000;
}
export-profile export-params {
  local-address 192.0.2.3;
  local-port 21111;
  dscp 20;
  forwarding-class assured-forwarding;
  loss-priority high;
  reporting-rate 20;
  format gpb;
  transport udp;
}
sensor interface-1 {
  server-name telemetry-server;
  export-name export-params;
  resource /junos/system/linecard/interface/logical/usage/;
  resource-filter et-*;
}
```

After you commit the configuration, verify that the sensor is enabled by issuing the **show agent sensors** operational command.

```
user@host> show agent sensors
```

Sensor Information :

Name	: interface-1
Resource	: /junos/system/linecard/interface/logical/usage/
Version	: 1.0
Sensor-id	: 193570469
Resource-filter	: et-*

Server Information :

Name	: telemetry-server
Scope-id	: 0
Remote-Address	: 192.0.2.2
Remote-port	: 30000

Profile Information :

Name	: export-params
Rep-interval	: 300
Address	: 192.0.2.3
Port	: 21111
Timestamp	: 1
Format	: GPB
Transport	: UDP
DSCP	: 20
Forwarding-class	: assured-forwarding
Loss-priority	: high

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are supported.
17.3R1	Starting with Junos OS Release 17.3R1, EX9200 switches, and the Routing and Control Board (RCB) on PTX3000 routers are also supported.
17.3R1	Starting with Junos OS Release 17.3R1 on MX Series routers only, you can specify a packet loss priority for an export profile.
17.2R1	Starting with Junos OS Release 17.2R1, QFX10000 and PTX1000 switches are also supported.
16.1R3	Starting with Junos OS Release 16.1R3, FPC1 and FPC2 on PTX Series routers are also supported.
15.1F5	Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.
15.1F3	Junos Telemetry Interface was introduced in Junos OS Release 15.1F3 on MX Series routers with interfaces configured on MPC1 through MPC6E and on PTX Series routers with interfaces configured on FPC3.

CHAPTER 4

Junos Telemetry Interface Configuration Statements and Operational Commands

- [export-profile \(Junos Telemetry Interface\) on page 26](#)
- [per-interface-per-member-link on page 30](#)
- [per-sid on page 31](#)
- [sensor \(Junos Telemetry Interface\) on page 32](#)
- [sensor-based-stats \(Junos Telemetry Interface\) on page 39](#)
- [streaming-server \(Junos Telemetry Interface\) on page 40](#)
- [show agent sensors](#)

export-profile (Junos Telemetry Interface)

Syntax	<pre>export-profile name { dscp value; format file-format; forwarding-class (assured-forwarding best-effort expedited-forwarding network-control); local-address ip-address; local-port source-port-number; loss-priority (high low medium-high medium-low); <payload-size bytes>; reporting-rate seconds; transport protocol-name; }</pre>
Hierarchy Level	[edit services analytics]
Release Information	<p>Statement introduced in Junos OS Release 15.1F3.</p> <p>payload-size bytes option introduced in Junos OS Release 16.1R3.</p> <p>Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers</p> <p>loss-priority option introduced in Junos OS Release 17.3R1 for MX Series routers only.</p> <p>Statement introduced in Junos OS Release 17.3R1 for EX9200 switches and the Routing and Control Board (RCB) on PTX3000 routers.</p> <p>Statement introduced in Junos OS Release 17.4R1 for virtual MX Series (vMX) routers.</p>
Description	<p>Configure the parameters of the export process for data generated through Junos Telemetry Interface sensors. You can create one or more export profiles. Each profile can be associated with one or more sensors that define the system resource to monitor and stream data. You can associate only one export profile with a specific sensor configuration.</p> <p>The IP layer delivers the exported data to the remote server. The export profile configuration allows you to specify a format for exported data, a transport protocol, the rate which the system generates data, and the local source port and IP address that are used to define the transport headers in the exported packets.</p> <p>To enable Junos Telemetry Interface, you must also configure a sensor that defines the parameters of the system resource to monitor and stream data, and a server to collect the data. To configure a sensor, include the sensor sensor-name statement at the [edit services analytics] hierarchy level. To configure the server that functions as a data collector, include streaming-server server-name statement at the [edit services analytics] hierarchy level.</p>



NOTE: Junos Telemetry Interface was introduced in Junos OS Release 15.1F3 on MX Series routers with interfaces configured on MPC1 through MPC6E and on PTX Series routers with interfaces configured on FPC3. Starting in

Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1 and FPC2 on PTX Series routers are also supported.

Starting with Junos OS Release 17.2R1, QFX10000 switches and PTX1000 routers are also supported.

Options *name*—Name of export profile.



NOTE: To associate this export profile with a configured sensor, include the name you configure for the export-profile statement at the [edit services analytics sensor *sensor-name* export-name] hierarchy level.

dscp value—Specify the DSCP value for the exported packets.

Range: 0 through 63.

Default: 0



NOTE: Any interface-level DSCP rewrite rules you have configured override the DSCP value you specify for the export profile. You need to specify a DSCP value for the export profile only if you do not configure DSCP rewrite rules on the outgoing interface. For more information, see *Configuring Rewrite Rules*.

format gpb—Specify the format to define the structure of exported data.

gpb—Google protocol buffers format.

forwarding-class (assured-forwarding | best-effort | expedited-forwarding | network-control)—(Packet Forwarding Engine sensors only) Specify the forwarding class for exported packets.

Default: best-effort

loss-priority (high | low | medium-high | medium-low) (MX Series only)—Specify the loss priority for exported packets. Loss priority settings help determine which packets are dropped from the network during periods of congestion.

local-address ip-address—Specify the source address of exported packets.

local-port number—Specify the source port for the exported packets.

payload-size bytes (Optional) —Specify the maximum size of exported packets.



NOTE:

The payload-size option is supported only on the following sensors:

- /junos/system/linecard/interface/
- /junos/system/linecard/interface/logical/usage/
- /junos/system/linecard/firewall/

Default: 5000 bytes.

Range: 2000 through 9192 bytes.



NOTE: Junos Telemetry Interface does not export packets larger than 9192 bytes.

reporting-rate *seconds*—Specify the interval at which the Junos Telemetry Interface sensor generates data to export to the collector.

As the configured interval expires, the most recent sample collected by the sensor is gathered and forwarded to the server configured to collect data.



NOTE: For Packet Forwarding Engine sensors, the minimum reporting rate is 2 seconds.

Range: 1 through 3600 (1 hour)

transport *protocol-name*—Specify the transport protocol to use to carry the telemetry data in the IP packets.

udp—User Datagram Protocol.

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

Related Documentation	• sensor on page 32
------------------------------	-------------------------------------

per-interface-per-member-link

Syntax	<code>per-interface-per-member-link (egress <i>egress-interface</i> ingress <i>ingress-interface</i>);</code>
Hierarchy Level	[edit protocols isis source-packet-routing sensor-based-stats],
Release Information	Statement introduced in Junos OS Release 17.4R1 on MX Series routers.
Description	<p>Configure sensor-based statistics per interface.</p> <p>Sensor-based statistics is the traffic statistics in a segment routing (SR) network that can be recorded in an OpenConfig compliant format for Layer 3 interfaces. The statistics is recorded for the Source Packet Routing in Networking (SPRING) traffic only, excluding RSVP and LDP-signaled traffic, and the family MPLS statistics per interface is accounted for separately. The SR statistics also includes SPRING traffic statistics per link aggregation group (LAG) member, and per segment identifier (SID).</p>
Options	<p><code>egress <i>egress-interface</i></code>—Enable sensor based statistics on the egress interface.</p> <p><code>ingress <i>ingress-interface</i></code>—Enable sensor based statistics on the ingress interface.</p>
Required Privilege Level	routing
Related Documentation	<ul style="list-style-type: none">• <i>Understanding Source Packet Routing in Networking (SPRING)</i>• <i>sensor-based-stats</i>• per-sid on page 31

per-sid

Syntax	<code>per-sid ingress <i>ingress</i>;</code>
Hierarchy Level	[edit protocols isis source-packet-routing sensor-based-stats],
Release Information	Statement introduced in Junos OS Release 17.4R1 on MX Series routers.
Description	<p>Configure sensor based statistics per Source Packet Routing in Networking (SPRING) route.</p> <p>Sensor-based statistics is the traffic statistics in a segment routing (SR) network that can be recorded in an OpenConfig compliant format for Layer 3 interfaces. The statistics is recorded for SPRING traffic only, excluding RSVP and LDP-signaled traffic, and the family MPLS statistics per interface is accounted for separately. The SR statistics also includes SPRING traffic statistics per link aggregation group (LAG) member, and per segment identifier (SID).</p>
Options	ingress <i>ingress</i> —Enable sensor based statistics for per-sid ingress accounting.
Required Privilege Level	routing
Related Documentation	<ul style="list-style-type: none">• <i>Understanding Source Packet Routing in Networking (SPRING)</i>• per-interface-per-member-link on page 30• <i>sensor-based-stats</i>

sensor (Junos Telemetry Interface)

Syntax	<pre>sensor <i>sensor-name</i> { export-name <i>export-profile-name</i>; resource <i>resource-string</i>; <resource-filter <i>regular expression</i>>; server-name [<i>streaming-server-names</i>]; }</pre>
Hierarchy Level	[edit services analytics]
Release Information	<p>Statement introduced in Junos OS Release 15.1F3.</p> <p>Support for MPC7E, MPC8E, and MPC9E on MX Series routers added in Junos OS Release 15.1F5.</p> <p>Support for FPC1 and FPC2 on PTX Series routers added in Junos OS Release 16.1R3.</p> <p>Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers.</p> <p>Statement introduced in Junos OS Release 17.3R1 for the Routing and Control Board (RCB) on PTX3000 routers, EX9200 switches, and MX150 routers.</p> <p>Statement introduced in Junos OS Release 17.4R1 for virtual MX series (vMX) routers.</p>
Description	<p>Configure a Junos Telemetry Interface sensor, which defines the parameters of a system resource to monitor and stream data. You can configure more than one sensor to stream data for the same system resource. For example, you might want to configure different parameters for exporting data for the same system resource. Additionally, you can use regular expressions to filter the data collected. Examples include filters for logical and physical interfaces and LSP messages. To apply different filters to the same system resource, you configure multiple sensors. For example, you can configure multiple logical interface sensors and apply a different interface filter to each one.</p>
Options	<p>Each sensor configuration requires you to specify the following: sensor name, an export profile name, a resource identifier string that enables monitoring and streaming of data for the specified system resource, and a server name to collect data. A regular expression to filter data for the specified resource is optional.</p> <p><i>sensor-name</i>—Specify a name that defines the sensor configuration. For example, for a sensor configuration that monitors all LSP events, you might choose the name lsp-mon-global. For a sensor configuration that monitors events only for an LSP named A2B, you might choose the name lsp-mon-A2B.</p> <p><i>export-name export-profile-name</i>—Specify the name of an export profile that you configured at the [edit services analytics export-profile name] hierarchy level to associate with the sensor. This export profile defines the parameters for exporting telemetry data, such as a format for exported data and the rate at which data is generated for export.</p>



NOTE: You can apply only one export profile to each sensor configuration.

The only supported transport protocol when you configure a sensor through the CLI is UDP.

resource *resource-string*—Enable the system resource to monitor and stream data. Each string corresponds to a specific system resource. The format is a file path and must be entered exactly. You can associate only one ***resource-string*** with a ***sensor-name***. Configure a separate sensor for each system resource you want to monitor. The resource string to enable LSP monitoring can be modified to specify a specific LSP.



NOTE: You can configure more than one sensor to monitor the same system resource. Configuring different sensors for the same system resource allows you configure different parameters for monitoring that resource.

Table 4 on page 34 lists each supported ***resource-identifier-string***, a description of the system resource monitored, and additional configuration information.

Table 4: resource statement Options

resource string	Description	Release Information
/junos/services/label-switched-path/usage/	<p>Packet Forwarding Engine sensor for LSP statistics. Starting with Junos OS Release 17.4R1 on MX Series and PTX Series routers only, statistics for bypass LSPs are also exported. Previously, only statistics for ingress LSPs were exported.</p> <p>For bypass LSPs, the following are exported:</p> <ul style="list-style-type: none"> • Bypass LSP originating at the ingress router of the protected LSP • Bypass LSP originating at the transit router of the protected LSP • Bypass LSP protecting the transit LSP as well as the locally originated LSP <p>When the bypass LSP is active, traffic is exported both on the bypass LSP and the ingress (protected) LSP.</p> <p>On MX Series routers only, bidirectional LSPs for ultimate-hop popping (UHP) are also supported.</p> <p>NOTE: You can modify <code>/junos/services/label-switched-path/usage/</code> to specify a specific LSP. Add <code>__instance__/lsp-name</code> to the end of the resource string identifier. For example, to monitor and stream data for LSP statistics for an LSP named mirror-to-murano-1, enter the following: <code>/junos/services/label-switched-path/usage/__instance__/mirror-to-murano-1</code>. If you do not specify a specific LSP name, the system resource monitors and streams data for all LSPs.</p> <p>When you enable a sensor for LSP statistics, you must also configure the <code>sensor-based-stats</code> statement at the <code>[edit protocols mpls]</code> hierarchy level. MX Series routers must also operate in enhanced mode. If not enabled by default, configure either the <code>enhanced-ip</code> statement or the <code>enhanced-ethernet</code> statement at the <code>[edit chassis network-services]</code> hierarchy level.</p>	<p>Junos OS Release 15.1F6 and later.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers.</p> <p>Junos OS Release 17.3 and later on EX9200 switches.</p>
/junos/system/linecard/cpu/memory/	Packet Forwarding Engine sensor for CPU memory.	<p>Junos OS Release 16.1R3 and later.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches.</p>
/junos/system/linecard/firewall/	Packet Forwarding Engine sensor for firewall filter counters and policer counters. Each line card reports counters separately.	<p>Junos OS Release 15.1F5 and later.</p> <p>Junos OS Release</p>

Table 4: resource statement Options (*continued*)

resource string	Description	Release Information
	<p>NOTE: Hierarchical policer statistics are collected for MX Series routers only. Traffic-class counter statistics are collected for PTX Series routers and QFX10000 switches only.</p> <p>Firewall counters are exported even if the interface to which the firewall filter is attached is down.</p>	<p>17.2R1 and later on QFX10000 switches.</p> <p>Junos OS Release 17.3R1 and later on PTX1000 routers and EX9200 switches.</p>
/junos/system/linecard/interface/	<p>Packet Forwarding Engine sensor for physical interface traffic.</p> <p>NOTE: For PTX Series routers, for a specific interface, queue statistics are exported for each line card. For MX series routers, interface queue statistics are exported only from the slot on which an interface is configured.</p> <p>For Aggregated Ethernet interfaces, statistics are exported for the member physical interfaces. You must aggregate the counters at the destination server, or collector.</p> <p>If a physical interface is administratively down or operationally down, interface counters are not exported.</p> <p>Issuing an operational clear command, such as clear interfaces statistics all, does not reset statistics exported by the line card.</p>	<p>Junos OS Release 15.1F3 and later on PTX Series routers only. Support introduced for MX Series routers in Junos OS Release 15.1F5.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches and MX150 routers.</p>
/junos/system/linecard/interface/logical/usage/	<p>Packet Forwarding Engine sensor for logical interface traffic.</p> <p>NOTE: If a logical interface is operationally down, interface statistics continue to be exported.</p> <p>Issuing an operational clear command, such as clear interfaces statistics all, does not reset statistics exported by the line card.</p> <p>NOTE: Locally injected packets from the Routing Engine are not exported.</p>	<p>Junos OS Release 15.1F5 and later.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches</p>
/junos/system/linecard/npu/memory/	<p>Packet Forwarding Engine sensor for network processing unit (NPU) memory.</p>	<p>Junos OS Release 16.1R3 and later.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches.</p>
/junos/system/linecard/npu/utilization/	<p>Packet Forwarding Engine sensor for NPU memory utilization and total memory available for each memory type.</p>	

Table 4: resource statement Options (*continued*)

resource string	Description	Release Information
		<p>Junos OS Release 16.1R3 and later.</p> <p>Junos OS Release 17.2R1 and later on QFX10000 switches.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches.</p>
/junos/npu-memory/	<p>Sensor that exports both NPU memory statistics from the Packet Forwarding Engine and flow-label statistics from the Routing Engine.</p> <p>To export only flow-label statistics, include the junos/npu-memory/flabel-memory/ resource string.</p> <p>To export only Packet Forwarding Engine NPU memory statistics, include /junos/system/linecard/npu/memory/ resource string.</p>	<p>Junos OS Release 16.1R3 and later on PTX Series routers only.</p> <p>NOTE: Junos OS Release 17.2R1 and later on PTX1000 routers.</p>
/junos/system/linecard/services/inline-jflow/	Packet Forwarding Engine sensor for performance metrics of the inline flow sampling process, such as the number of active flows and the number of exported flows.	<p>Junos OS Release 16.1R3 and later on MX series and PTX series routers only.</p> <p>Junos OS Release and later on EX9200 switches, PTX1000 routers, and MX150 routers.</p>
/junos/system/linecard/optics/	Packet Forwarding Engine sensor for various optical performance metrics, such as transmit and receive power levels.	<p>Junos OS Release 16.1R3 and later.</p> <p>Junos OS Release and later 17.2R1 on QFX10000 switches.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches and PTX1000 routers.</p>
/junos/system/linecard/qmon/	<p>Sensor for queue depth statistics for ingress and egress queue traffic. Statistics are exported directly from the line card.</p> <p>NOTE: Issuing an operational clear command, such as clear interfaces statistics all, does not reset the statistics exported by the line card.</p>	

Table 4: resource statement Options (*continued*)

resource string	Description	Release Information
		<p>Junos OS Release 17.1R1 and later on MX Series routers on MPC7E, MPC8E, and MPC9E only.</p> <p>Junos OS 17.3R1 and later on EX9200 switches.</p> <p>NOTE: virtual MX Series (vMX) routers are not supported.</p>
/junos/system/linecard/fabric/	<p>Sensor for fabric statistics.</p> <p>The following types of statistics can be exported:</p> <ul style="list-style-type: none"> Fabric statistics for Packet Forwarding Engine pairs (resource-filter option is not supported) FPC fabric statistics Control Board and Switch Fabric Board fabric statistics. 	<p>Junos OS Release 17.2R1 and later on MX Series routers only.</p> <p>Junos OS Release 17.3R1 and later on EX9200 switches.</p> <p>NOTE: virtual MX Series (vMX) routers are not supported.</p>
/junos/system/linecard/packet/usage/	Sensor for Packet Forwarding Engine Statistics. This sensor exports statistics for counters and provides visibility into Packet Forwarding Engine error and drop statistics.	Junos OS Release 17.4R1 and later on MX Series and PX Series routers.
/junos/services/segment-routing/interface/ingress/usage/ /junos/services/segment-routing/interface/egress/usage/ /junos/services/segment-routing/sid/usage/	<p>Sensors for aggregate segment routing traffic with IS-IS.</p> <p>The first path exports inbound traffic. The second path exports outbound traffic. The third path exports inbound segment routing traffic for each segment identifier.</p> <p>NOTE: When you enable a sensor for segment routing statistics, you must also configure the sensor-based-stats statement at the [edit protocols isis source-packet-routing] hierarchy level. MX Series and PTX Series routers must also operate in enhanced mode. On MX Series routers, if not enabled by default, configure either the enhanced-ip statement or the enhanced-ethernet statement at the [edit chassis network-services] hierarchy level. On PTX Series routers, configure the enhanced-mode statement at the [edit chassis network-services] hierarchy level.</p>	Junos OS Release 17.4 and later on MX Series and PTX5000 routers.

resource-filter *regular-expression*—(Optional) Specify a regular expression to filter data for a specific resource. For example, you can filter for a specific set of logical or physical interfaces, firewall filters, or LSP messages. When you configure a system resource to monitor and stream data globally—that is, systemwide—you do not need to include a regular expression.

Examples of regular expressions to filter data exported through sensor configuration:

- Logical interface statistics sensor—`et-2/0/7:1*`
- LSP events sensor—`lsp-from-A-to-B*`
- Firewall filter counters sensor—`f_test1*`

server-name [*streaming- server-names*]—Specify one or more servers to transport data for collection. Include at least one server-name configured at the **[edit services analytics *streaming-server* server-name]** hierarchy level.




NOTE: Starting in Junos OS Release 15.1F6, you can configure as many as four streaming servers for a single sensor configuration. In previous releases, you can specify only one streaming server for each configured sensor. To specify more than one streaming server for a sensor, you must enclose the names in brackets.

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

Related Documentation	<ul style="list-style-type: none"> • export-profile on page 26
------------------------------	---

sensor-based-stats (Junos Telemetry Interface)

Syntax	sensor-based-stats;
Hierarchy Level	[edit protocols mpls]
Syntax	<pre>sensor-based stats { per-interface-per-member-link (ingress <i>interface-name</i> egress <i>interface-name</i>); per-sid ingress <i>interface-name</i>; }</pre>
Hierarchy Level	[edit protocols isis source-packet-routing]
Release Information	<p>Statement introduced in Junos OS Release 15.1F6.</p> <p>Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers.</p> <p>Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.</p> <p>The IS-IS hierarchy and the per-interface-per-member-link and per-sid options introduced in Junos OS Release 17.4R1 for MX Series routers and PTX5000 routers.</p>
Description	<p>For the MPLS hierarchy, enable the collection of LSP statistics for the Junos Telemetry Interface. You must configure this statement when you configure a sensor to monitor and stream data for LSP statistics. To enable a sensor to stream data for LSP statistics through UDP, include the resource /junos/services/label-switched-path/usage/ statement at the [edit services analytics sensor <i>sensor-name</i>] hierarchy level.</p> <p>For additional information about configuring an LSP statistics sensor to stream data through gRPC, see “Guidelines for gRPC Sensors (Junos Telemetry Interface)” on page 70.</p> <p>For the IS-IS hierarchy, enable the collection of aggregate segment routing statistics.</p>
	<p>.....</p> <div>  <p>NOTE: Only MX Series routers and PTX5000 routers support this hierarchy.</p> </div> <p>.....</p>
Options	The remaining options are explained separately.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Understanding the Junos Telemetry Interface Export Format of Collected Data on page 10

streaming-server (Junos Telemetry Interface)

Syntax `streaming-server streaming-server-name {
 remote-address ip-address;
 remote-port number;
 }`

Hierarchy Level `[edit services analytics]`

Release Information Statement introduced in Junos OS Release 15.1F3.
Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers.
Statement introduced in Junos OS Release 17.3R1 for the Routing and Control Board (RCB) on PTX3000 routers and EX9200 switches.
Statement introduced in Junos OS Release 17.4R1 for virtual MX Series (vMX) routers.

Description For Junos Telemetry Interface, configure the parameters of the server that collects exported data streamed by a monitored system resource. You can configure more than one streaming server. To collect data, you must associate a configured server with one or more configured sensors. The sensor configuration defines the parameters to monitor a specific system resource. To configure a sensor, include the `sensor sensor-name` statement at the `[edit services analytics]` hierarchy level.

To configure the server that collects data, you must also configure a destination IP address and a destination port. Junos Telemetry Interface relies on neighbor reachability information to deliver packets to the destination address. That means that all policies, such as filtering, that apply to the packets for that destination also apply to the exported packets.



NOTE: Starting with Junos OS Release 15.1F6, you can also associate more than one server with a specific sensor configuration, which enables you to transmit streamed data for the same sensor to more than one server.



NOTE: Junos Telemetry Interface was introduced in Junos OS Release 15.1F3 on MX Series routers with interfaces configured on MPC1 through MPC6E and on PTX Series routers with interfaces configured on FPC3. Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1 and FPC2 on PTX Series routers are also supported.

Options ***streaming-server-name***—Specify a name for the server configured to collect data streamed through Junos Telemetry Interface. You can configure multiple streaming servers. To associate as many as four server names with a sensor configuration, include each name at the **[edit services analytics sensor *sensor-name* streaming server [*streaming-server-names*]]** hierarchy level. If you specify more than one streaming server, you must enclose the names in brackets.

remote-address ***ip-address***—Specify the destination address of the streaming server for exported packets.

remote-port ***number***—Specify a port number for the destination address of the streaming server for exported packets.

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

Related Documentation • [export-profile \(Junos Telemetry Interface\) on page 26](#)

show agent sensors

Syntax `show agent sensors`

Release Information Statement introduced in Junos OS Release 15.1F3
Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches, QFX5200 switches, and PTX1000 routers in Junos OS Release 17.2R1.
Statement introduced in Junos OS Release 17.3R1 for QFX5110 switches, EX9200 switches and the Routing and Control Board (RCB) on PTX3000 routers.

Description Display information about sensors configured for Junos Telemetry Interface.



NOTE: Junos Telemetry Interface was introduced in Junos OS Release 15.1F3 on MX Series routers with interfaces configured on MPC1 through MPC6E and on PTX Series routers with interfaces configured on FPC3. Starting in Junos OS Release 15.1F5, Junos Telemetry Interface is also supported on MPC7E, MPC8E, and MPC9E on MX Series routers.

Starting with Junos OS Release 16.1R3, FPC1, FPC2, and dual Routing Engines on PTX Series routers are also supported.

Required Privilege Level view

Related Documentation

- [export-profile on page 26](#)
- [sensor on page 32](#)
- [streaming-server on page 40](#)

List of Sample Output [show agent sensors \(firewall filter sensor\) on page 43](#)
[show agent sensors \(CPU memory sensor\) on page 44](#)

Output Fields [Table 5 on page 42](#) lists the output fields for the **show agent sensors** command. Output fields are listed in the approximate order in which they appear.

Table 5: show agent sensors Output Fields

Field Name	Field Description
Sensor Information	Information about sensors configured to monitor system resources and stream data.
Name	Name of configured sensor.
Resource	Resource string used to configure and identify the system resource enabled to monitor and stream data..

Table 5: show agent sensors Output Fields (*continued*)

Field Name	Field Description
Sensor-id	Numerical identifier of the sensor.
Server Information	Information about servers configured to collect sensor data.
Name	Name of server.
Scope-id	
Remote-Address	Destination IP address for exported packets.
Remote-port	Destination port for exported packets.
Profile information	Information about export profiles for sensors.
Name	Name of export profile.
Rep-interval	Interval, in seconds, at which the sensor generates data to export.
Address	Source address of exported packets.
Port	Source port of exported packets.
Format	Format of exported data message: GPB
DSCP	Configured DSCP value for exported packets. NOTE: The default value is 0. This value is displayed if you do not configure a DSCP value.
Forwarding-class	Configured forwarding class for exported packets. NOTE: The default value is 0. This value is displayed if you do not configure a forwarding class.
Loss-Priority	Configured loss priority for packets streamed through UDP (MX Series only): high, low, medium-high, medium-low

Sample Output

show agent sensors (firewall filter sensor)

```

user@host> show agent sensors
Sensor Information :

    Name                :firewall-stats
    Resource             :/junos/system/linecard/firewall/
    Sensor ID            :93390914

Server Information :

    Name                :jvision-server

```

```
Scope ID           :0
Remote-Address     :160.1.1.1
Remote-port        :2001
```

Profile Information :

```
Name               :export-common
Rep-interval       :2
Address            :160.1.1.2
Port               :1000
Timestamp          :1
Format             :GPB
Transport          :UDP
DSCP               :0
Forwarding-class   :0
Loss-priority      :high
```

show agent sensors (CPU memory sensor)

```
user@host> show agent sensors
```

Sensor Information :

```
Name               : se1
Resource           :/junos/system/cpu/memory/
Version            : 1.0
Sensor-id          : 114833
Subscription-ID    : 562949953536145
Parent-Sensor-Name : Not applicable
Component(s)       : PFE
```

Server Information :

```
Name               : ser1
Scope-id           : 0
Remote-Address     : 10.3.3.3
Remote-port        : 6000
Transport-protocol : UDP
```

Profile Information :

```
Name               : ex1
Reporting-interval : 1
Payload-size       : 5000
Address            : 0.0.0.0
Port               : 1000
Timestamp          : 1
Format             : GPB
DSCP               : 0
Forwarding-class   : assured-forwarding
Loss-priority      : high
```

CHAPTER 5

Decoding Data

- [Decoding Junos Telemetry Interface Data With UNIX Utilities on page 45](#)

Decoding Junos Telemetry Interface Data With UNIX Utilities

You can use UNIX utilities to decode Junos Telemetry Interface data on a server, or collector, that is streaming data from a Juniper Networks device. The example in this section shows you how to decode a single packet of streamed data.

Preparing the Collector to Decode Data

This example requires the following:

- UNIX OS with the Netcat (nc) utility.
- Protocol buffers compiler.
- Junos Telemetry Interface protocol buffers files.

This procedure shows how to prepare the collector to decode data using the Ubuntu OS.

1. Install the Netcat utility.

```
sudo apt-get install netcat
```

2. Install the protocol buffers compiler.

```
sudo apt-get install protobuf-compiler
```

3. Install the protocol buffers developer's library.

```
sudo apt-get install libprotobuf-dev
```

4. Verify that the library files are installed.

```
ls /usr/include/google/protobuf/descriptor.proto  
/usr/include/google/protobuf/descriptor.proto
```

5. Download and install the latest version of the Junos Telemetry interface protocol buffers files.

From a Web browser, navigate to the All Junos Platforms software download URL on the Juniper Networks page: <http://www.juniper.net/support/downloads/>. After you select the name of the Junos OS platform and the release number, go to the **Tools** section and download the **Junos Telemetry Interface Data Model Files** package.

```
tar -xvzf junos-telemetry-interface-15.1F6.9.tgz
junos-telemetry-interface/telemetry_top.proto
junos-telemetry-interface/logical_port.proto
junos-telemetry-interface/lsp_mon.proto
junos-telemetry-interface/firewall.proto
junos-telemetry-interface/lsp_stats.proto
junos-telemetry-interface/port.proto
junos-telemetry-interface/NOTICE
junos-telemetry-interface/license.txt
```



NOTE: Be sure to note the location of the extracted files.

Decoding Data on the Collector

This procedure shows you how to capture data, decode raw data, and use the protocol buffers files to decode data.

To decode data:

1. Capture the data.

Run netcat on a destination streaming telemetry server, or collector, in UDP listener mode to store all incoming datagrams into a file. Use the destination port number configured in streaming-server profile on your Juniper Networks device.

```
nc -ul 0.0.0.0 20000 > data.gpb
```



NOTE: This command stores datagrams into a file named **data.gpb**. Run this program to capture data. When you want to stop receiving data, stop with the program by sending the break signal (Control + C)

2. Decode raw data.



NOTE: This step is optional. It is not required if you know the encoded message type of the data.

Decode the message from the **data.gpb** file.

```
protoc --decode_raw < ../data.gpb
1: "hillrock:160.1.1.25"
2: 0
4:
"S1:/junos/system/linecard/interface/logical/usage:/junos/system/linecard/interface/logical/usage/:PFE"
5: 65265
```

```

6: 1477686534474
7: 1
8: 1
101 {
  2636 {
    7 {
      1 {
        1: "et-0/0/4:2.32767"
        2: 1477642750
        3: 813
        4 {
          12: 0x37363732332e3165
        }
      }
    }
  }
}
.
.
.

```

The next nested structure under **2636** identifies the sensor type. The numerical value **2636** identifies the **JuniperNetworksSensor** message, which is defined in the **telemetry_top.proto** file. In this example, the numerical identifier **7** corresponds to the **LogicalPort** message defined in the **logical_port.proto** file. Use this information in the next step to generate more detailed output.

3. Decode the message to include field names.

Run the protocol buffers compiler with the decode option. Additionally, specify the top-level message type (**TelemetryStream**) and the file with the message definition, **logical_port.proto**. You must also include the Goggle protocol buffers (gpb) library.

```

protoc --decode TelemetryStream logical_port.proto -I /usr/include -I . <
data.gpb
system_id: "hillrock:160.1.1.25"
component_id: 0
sensor_name:
"SL:/junos/system/linecard/interface/logical/usage:/junos/system/linecard/interface/logical/usage:/PFE"
sequence_number: 65268
timestamp: 1477686536484
version_major: 1
version_minor: 1
enterprise {
  [juniperNetworks] {
    [jnprLogicalInterfaceExt] {
      interface_info {
        if_name: "et-0/0/4:2.32767"
        init_time: 1477642750
        snmp_if_index: 813
        parent_ae_name: "ae1.32767"
        ingress_stats {
          if_packets: 0
          if_octets: 0
        }
        egress_stats {
          if_packets: 0
          if_octets: 0
        }
        op_state {
          operational_status: "up"
        }
      }
    }
  }
}

```

```
    }
  }
  interface_info {
    if_name: "et-0/0/7:3.0"
    init_time: 1477642750
    snmp_if_index: 520
    parent_ae_name: "ae0.0"
    ingress_stats {
      if_packets: 61203309
      if_octets: 6487548454
    }
    egress_stats {
      if_packets: 87416547
      if_octets: 9266153982
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.0"
    init_time: 1477642750
    snmp_if_index: 2512
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214806
    }
    egress_stats {
      if_packets: 26247215
      if_octets: 2781829290
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.1"
    init_time: 1477642750
    snmp_if_index: 2522
    ingress_stats {
      if_packets: 26266249
      if_octets: 2784214972
    }
    egress_stats {
      if_packets: 26249115
      if_octets: 2781935590
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.2"
    init_time: 1477642750
    snmp_if_index: 2523
    ingress_stats {
      if_packets: 26266248
      if_octets: 2784214912
    }
    egress_stats {
      if_packets: 26249106
```

```
        if_octets: 2781935086
      }
      op_state {
        operational_status: "up"
      }
    }
  interface_info {
    if_name: "et-0/0/13:0.3"
    init_time: 1477642750
    snmp_if_index: 2524
    ingress_stats {
      if_packets: 26266248
      if_octets: 2784214820
    }
    egress_stats {
      if_packets: 26248520
      if_octets: 2781902320
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.4"
    init_time: 1477642750
    snmp_if_index: 2525
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214760
    }
    egress_stats {
      if_packets: 26247302
      if_octets: 2781834112
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.5"
    init_time: 1477642750
    snmp_if_index: 2526
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214760
    }
    egress_stats {
      if_packets: 26247209
      if_octets: 2781828904
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.6"
    init_time: 1477642750
    snmp_if_index: 2527
    ingress_stats {
      if_packets: 26266248
      if_octets: 2784214820
    }
```

```
    }
    egress_stats {
      if_packets: 26247196
      if_octets: 2781828226
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.7"
    init_time: 1477642750
    snmp_if_index: 2528
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214760
    }
    egress_stats {
      if_packets: 26247203
      if_octets: 2781828618
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.8"
    init_time: 1477642750
    snmp_if_index: 2529
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214760
    }
    egress_stats {
      if_packets: 26247225
      if_octets: 2781829850
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.9"
    init_time: 1477642750
    snmp_if_index: 2530
    ingress_stats {
      if_packets: 26266247
      if_octets: 2784214760
    }
    egress_stats {
      if_packets: 26247209
      if_octets: 2781828954
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.32767"
    init_time: 1477642750
    snmp_if_index: 648
```



```
    ingress_stats {
      if_packets: 4
      if_octets: 240
    }
    egress_stats {
      if_packets: 0
      if_octets: 0
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/4:2.32767"
    init_time: 1477642750
    snmp_if_index: 813
    parent_ae_name: "ae1.32767"
    ingress_stats {
      if_packets: 0
      if_octets: 0
    }
    egress_stats {
      if_packets: 0
      if_octets: 0
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/7:3.0"
    init_time: 1477642750
    snmp_if_index: 520
    parent_ae_name: "ae0.0"
    ingress_stats {
      if_packets: 61206122
      if_octets: 6487846632
    }
    egress_stats {
      if_packets: 87420567
      if_octets: 9266580102
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.0"
    init_time: 1477642750
    snmp_if_index: 2512
    ingress_stats {
      if_packets: 26267458
      if_octets: 2784343172
    }
    egress_stats {
      if_packets: 26248420
      if_octets: 2781957020
    }
    op_state {
      operational_status: "up"
    }
  }
```

```
}
interface_info {
  if_name: "et-0/0/13:0.1"
  init_time: 1477642750
  snmp_if_index: 2522
  ingress_stats {
    if_packets: 26267460
    if_octets: 2784343338
  }
  egress_stats {
    if_packets: 26250320
    if_octets: 2782063320
  }
  op_state {
    operational_status: "up"
  }
}
interface_info {
  if_name: "et-0/0/13:0.2"
  init_time: 1477642750
  snmp_if_index: 2523
  ingress_stats {
    if_packets: 26267459
    if_octets: 2784343278
  }
  egress_stats {
    if_packets: 26250311
    if_octets: 2782062816
  }
  op_state {
    operational_status: "up"
  }
}
interface_info {
  if_name: "et-0/0/13:0.3"
  init_time: 1477642750
  snmp_if_index: 2524
  ingress_stats {
    if_packets: 26267460
    if_octets: 2784343292
  }
  egress_stats {
    if_packets: 26249725
    if_octets: 2782030050
  }
  op_state {
    operational_status: "up"
  }
}
interface_info {
  if_name: "et-0/0/13:0.4"
  init_time: 1477642750
  snmp_if_index: 2525
  ingress_stats {
    if_packets: 26267459
    if_octets: 2784343232
  }
  egress_stats {
    if_packets: 26248507
    if_octets: 2781961842
  }
}
```

```

        op_state {
            operational_status: "up"
        }
    }
    interface_info {
        if_name: "et-0/0/13:0.5"
        init_time: 1477642750
        snmp_if_index: 2526
        ingress_stats {
            if_packets: 26267459
            if_octets: 2784343232
        }
        egress_stats {
            if_packets: 26248414
            if_octets: 2781956634
        }
        op_state {
            operational_status: "up"
        }
    }
    interface_info {
        if_name: "et-0/0/13:0.6"
        init_time: 1477642750
        snmp_if_index: 2527
        ingress_stats {
            if_packets: 26267460
            if_octets: 2784343292
        }
        egress_stats {
            if_packets: 26248401
            if_octets: 2781955956
        }
        op_state {
            operational_status: "up"
        }
    }
    interface_info {
        if_name: "et-0/0/13:0.7"
        init_time: 1477642750
        snmp_if_index: 2528
        ingress_stats {
            if_packets: 26267459
            if_octets: 2784343232
        }
        egress_stats {
            if_packets: 26248408
            if_octets: 2781956348
        }
        op_state {
            operational_status: "up"
        }
    }
    interface_info {
        if_name: "et-0/0/13:0.8"
        init_time: 1477642750
        snmp_if_index: 2529
        ingress_stats {
            if_packets: 26267459
            if_octets: 2784343232
        }
        egress_stats {

```

```
        if_packets: 26248430
        if_octets: 2781957580
      }
      op_state {
        operational_status: "up"
      }
    }
  interface_info {
    if_name: "et-0/0/13:0.9"
    init_time: 1477642750
    snmp_if_index: 2530
    ingress_stats {
      if_packets: 26267459
      if_octets: 2784343232
    }
    egress_stats {
      if_packets: 26248414
      if_octets: 2781956684
    }
    op_state {
      operational_status: "up"
    }
  }
  interface_info {
    if_name: "et-0/0/13:0.32767"
    init_time: 1477642750
    snmp_if_index: 648
    ingress_stats {
      if_packets: 4
      if_octets: 240
    }
    egress_stats {
      if_packets: 0
      if_octets: 0
    }
    op_state {
      operational_status: "up"
    }
  }
}
}
```

Related Documentation

- [Configuring a Junos Telemetry Interface Sensor \(CLI Procedure\) on page 15](#)

PART 3

Configuring gRPC Sensors

- [OpenConfig and gRPC for Junos Telemetry Interface on page 57](#)

CHAPTER 6

OpenConfig and gRPC for Junos Telemetry Interface

- [Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 58](#)
- [Installing the Network Agent Package \(Junos Telemetry Interface\) on page 61](#)
- [gRPC Services for Junos Telemetry Interface on page 64](#)
- [ssl on page 69](#)
- [Guidelines for gRPC Sensors \(Junos Telemetry Interface\) on page 70](#)
- [Understanding YANG on Devices Running Junos OS on page 143](#)
- [Configurable NETCONF Proxy for Junos Telemetry Interface on page 144](#)
- [request system yang add](#)
- [request system yang delete](#)
- [request system yang update](#)
- [request system yang validate](#)

Understanding OpenConfig and gRPC on Junos Telemetry Interface

Starting in Junos OS Release 16.1R3, you can use a set of remote procedure call (RPC) interfaces to configure the Junos Telemetry Interface and stream telemetry data using the gRPC framework. OpenConfig supports the use of vendor-neutral data models for configuring and managing multivendor networks. gRPC is an open source framework that provides secure and reliable transport of data.



NOTE: OpenConfig for Junos OS and gRPC are supported only on MPCs on MX Series and on PTX Series routers starting with Junos OS Release 16.1R3.

Starting with Junos OS Release 17.2R1, OpenConfig and gRPC are also supported on QFX10000 switches, QFX5200 switches, and PTX1000 routers.

Starting with Junos OS Release 17.3R1, Junos Telemetry Interface is supported on the Routing Control and Board (RCB) on PTX3000 routers, QFX5110 switches, and EX4600 and EX9200 switches.

OpenConfig and gRPC are not supported on MX80 and MX104 routers.

- [Network Agent Software on page 58](#)
- [Using OpenConfig for Junos OS to Enable Junos Telemetry Interface on page 58](#)
- [Using gRPC to Stream Data on page 59](#)

Network Agent Software

Implementing OpenConfig with gRPC for Junos Telemetry Interface requires that you download and install a package called Network Agent if your Juniper Networks device is running a version of Junos OS with Upgraded FreeBSD. For all other versions of Junos OS, the Network Agent functionality is embedded in the software. Network Agent functions as a gRPC server and terminates the OpenConfig RPC interfaces. It is also responsible for streaming the telemetry data according to the OpenConfig specification. To view the OpenConfig specification for telemetry, see the [OpenConfig Telemetry specification](#). For more information about OpenConfig for Junos OS, see the *OpenConfig Feature Guide*.

The Network Agent component also supports server-based Secure Sockets Layer (SSL) authentication. Client-based SSL authentication is not supported. You must install SSL certificates on your Juniper Networks device.

For information about installing the Network Agent package, see [“Installing the Network Agent Package” on page 61](#).

Using OpenConfig for Junos OS to Enable Junos Telemetry Interface

OpenConfig for Junos OS specifies an RPC model to enable the Junos Telemetry Interface. You must download and install the OpenConfig for Junos OS package on your Juniper Networks device. This package also includes the required YANG models. Using a Web browser, navigate to the All Junos Platforms software download URL on the Juniper Networks webpage: <http://www.juniper.net/support/downloads/>. From the **Network**

Management tab, scroll down to select **OpenConfig**. Select the **Software** tab. Select the appropriate version of OpenConfig module. Two versions are available, one for devices running Junos OS with Upgraded FreeBSD and another for devices running all other versions of Junos OS. For more information, see *Installing the OpenConfig Package* and *Understanding Junos OS YANG Modules*.

The programmatic interface **OpenConfigTelemetry** that is installed by the Network Agent package defines the telemetry gRPC service. The **telemetrySubscribe** RPC specifies the following subscription parameters:

- OpenConfig path that identifies the system resource to stream telemetry data, for example:
`/interfaces/interface/state/counters/`
- Interval at which data is reported and streamed to the collector server, in milliseconds, for example:
`sample_frequency = 4000`

The **telemetrySubscribe** RPC is used by a streaming server, or collector, to request an inline subscription for data at the specified path. The device should then send telemetry data back on the same connection as the subscription request.

Using gRPC to Stream Data

Per the OpenConfig specification, only gRPC-based transport is supported for streaming data. The gRPC server that is installed by the Network Agent package terminates the gRPC sessions from the management system that runs the client. RPC calls trigger the creation of Junos OS sensors that either stream data periodically or report events, which are then funneled onto the appropriate gRPC channel by Network Agent.

See [Table 6 on page 59](#) for a list and descriptions of the RPCs implemented to the support the Junos Telemetry Interface.

Table 6: Telemetry RPCs

RPC Name	Description
telemetrySubscribe	Specify telemetry parameters and stream data for the specified list of OpenConfig paths.
getTelemetrySubscriptions	Retrieve the list of subscriptions that are created through telemetrySubscribe .
cancelSubscription	Unsubscribe a subscription created through telemetrySubscribe .

Data streamed through gRPC is formatted in OpenConfig key/value pairs in Google protocol buffers (gpb) messages. In this universal format, keys are strings that correspond to the path of the system resources in the OpenConfig schema for the device being monitored. The values correspond to integers or strings that identify the operational state of the system resource, such as interface counters, and the state of the resource.

The following shows the universal key/value format:

```
message KeyValue {
    string key          = 1 [(telemetry_options).is_key = true];
    uint64 int_value    = 2;
    string str_value    = 3;
    string prefix_str = 4;
}

message TelemetryStream {
    // router name or export IP address
    required string system_id      = 1 [(telemetry_options).is_key = true];

    // line card / RE (slot number)
    optional uint32 component_id  = 2 [(telemetry_options).is_key = true];

    // PFE (if applicable)
    optional uint32 sub_component_id = 3 [(telemetry_options).is_key = true];

    // timestamp (common to all entries in the kv array)
    optional uint64 timestamp      = 4 [(telemetry_options).is_timestamp = true];

    // key / value pairs
    repeated KeyValue kv;
}
```

The following example shows how a set of counters for an interface can be represented:

```
key = "/interfaces/counters/rx-bytes",    int_value = 1000
key = "/interfaces/counters/tx-bytes",    int_value = 2000
key = "/interfaces/counters/rx-packets",  int_value = 10
key = "/interfaces/counters/rx-bytes" ,   int_value = 20
key = "/interfaces/counters/oper-state",  str_value = "up"
```

The Network Agent package provides a mapping table that maps field names to the OpenConfig key strings.

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, Junos Telemetry Interface is supported on the Routing Control and Board (RCB) on PTX3000 routers, QFX5110 switches, and EX4600 and EX9200 switches.
17.2R1	Starting with Junos OS Release 17.2R1, OpenConfig and gRPC are also supported on QFX10000 switches, QFX5200 switches, and PTX1000 routers.
16.1R3	Starting in Junos OS Release 16.1R3, you can use a set of remote procedure call (RPC) interfaces to configure the Junos Telemetry Interface and stream telemetry data using the gRPC framework.
16.1R3	OpenConfig for Junos OS and gRPC are supported only on MPCs on MX Series and on PTX Series routers starting with Junos OS Release 16.1R3.

**Related
Documentation**

- [Installing the Network Agent Package \(Junos Telemetry Interface\) on page 61](#)
- [Release Information for Junos OS with Upgraded FreeBSD](#)

Installing the Network Agent Package (Junos Telemetry Interface)

Starting with Junos OS Release 16.1R3, the Junos Network Agent software package provides a framework to support OpenConfig and gRPC for the Junos Telemetry Interface on MX Series routers and PTX5000 routers. The Network Agent package functions as a gRPC server that terminates the OpenConfig remote procedure call (RPC) interfaces and streams the telemetry data according to the OpenConfig specification. The Junos Network Agent package, which runs on the Routing Engine, implements local statistics collection and reports data to active telemetry stream subscribers.

Starting with Junos OS Release 17.2R1, the Junos Network Agent Package is also supported on QFX10000 switches and QFX5200 switches.

Starting with Junos OS Release 17.3R1, the Junos Network Agent Package is supported on QFX5110 switches and EX9200 switches.

The Junos Network Agent is available as a separate package only for Junos OS with Upgraded FreeBSD. This package also includes the required YANG models. For other versions of Junos OS, Network Agent functionality is embedded in the software. For more information about Junos OS with Upgraded FreeBSD, see *Release Information for Junos OS with Upgraded FreeBSD*.

Network Agent for Junos OS software package has the following naming conventions:

- Package Name—This is **Network-Agent**.
- Architecture—This field indicates the CPU architecture of the platforms, such as **x86**.
- Application Binary Interface (ABI)—This field indicates the “word length” of the CPU architecture. Values include **32** for 32-bit architectures and **64** for 64-bit architectures.

- Release—This field indicates the Junos OS release number, such as **16.1R3.16**.
- Package release and spin number—This field indicates the package version and spin number, such as **C1.1**.

All Junos Network Agent packages are in tarred and gzipped (**.tgz**) format.



NOTE: Each version of the Network Agent package is supported on a single release of Junos OS only. The Junos OS version supported is identified by the Junos OS release number included in the Network Agent package name.

Examples of valid Network Agent package names including the following:

- **network-agent-x86-64-16.1R3.16-C1.0.tgz**
- **network-agent-x86-32-16.1R4.12-C1.1.tgz**

Before you begin:

- Install Junos OS Release 16.1R3 or later.
- Install the OpenConfig for Junos OS module. Using a Web browser, navigate to the All Junos Platforms software download URL on the Juniper Networks webpage: <http://www.juniper.net/support/downloads/>. From the **Network Management** tab, scroll down to select **OpenConfig**. Select the **Software** tab. Select the **OpenConfig Package (Junos with upgraded FreeBSD)**. For more information, see *Installing the OpenConfig Package*.
- Install Secure Sockets Layer (SSL) certificates of authentication on your Juniper Networks device.



NOTE: Only server-based SSL authentication is supported. Client-based authentication is not supported.

To download and install the Network Agent package:

1. Using a Web browser, navigate to the All Junos Platforms software download URL on the Juniper Networks webpage: <http://www.juniper.net/support/downloads/>.
2. Select the name of the Junos OS platform for the software that you want to download.
3. Select the release number (the number of the software version that you want to download) from the **Release** drop-down list to the right of the Download Software page.
4. Select the **Software** tab.

5. In the **Tools** section of the **Software** tab, select the **Junos Network Agent** package for the release.
6. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by a Juniper Networks representative.
7. Download the software to a local host.
8. Copy the software to Juniper Networks device or to your internal software distribution site.
9. Install the new **network-telemetry** package on the device by issuing the **request system software add package-name** from the operational mode:

For example:

```
user@host > request system software add
network-telemetry-x86-64-16.1R3.16-C1.0.tgz
```



NOTE: The command uses the **validate** option by default. This option validates the software package against the current configuration as a prerequisite to adding the software package to ensure that the device reboots successfully. This is the default behavior when the software package being added is a different release.

10. Issue the **show version | grep na\ telemetry** command to verify that the Network Agent package was successfully installed.

```
user@host> show version | grep na\ telemetry
JUNOS na telemetry
[20161109.201405_builder_junos_161_r3]
```

For information about configuring gRPC services on your Juniper Networks device, see [“gRPC Services for Junos Telemetry Interface” on page 64](#).

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, the Junos Network Agent Package is supported on QFX5110 switches and EX9200 switches.
17.2R1	Starting with Junos OS Release 17.2R1, the Junos Network Agent Package is also supported on QFX10000 switches and QFX5200 switches.
16.1R3	Starting with Junos OS Release 16.1R3, the Junos Network Agent software package provides a framework to support OpenConfig and gRPC for the Junos Telemetry Interface on MX Series routers and PTX5000 routers.

Related Documentation

- [Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 58](#)

gRPC Services for Junos Telemetry Interface

- [Configuring gRPC for the Junos Telemetry Interface on page 64](#)
- [Configuring Bidirectional Authentication for gRPC for Junos Telemetry Interface on page 66](#)

Configuring gRPC for the Junos Telemetry Interface

Starting with Junos OS Release 16.1R3 on MX Series routers and PTX3000 and PTX5000 routers, you can stream telemetry data for various network elements through gRPC, an open source framework for handling remote procedure calls based on TCP. The Junos Telemetry Interface relies on a so-called push model to deliver data asynchronously, which eliminates polling. For all Juniper devices that run a version of Junos OS with upgraded FreeBSD kernel, you must install the Junos Network Agent software package, which provides the interfaces to manage gRPC subscriptions. For Juniper Network devices that run other all other versions of the Junos OS, this functionality is embedded in the Junos OS software. For more information about installing the Junos Network Agent package, see [“Installing the Network Agent Package” on page 61](#).

The Junos Telemetry Interface and gRPC streaming are supported on QFX10000 and QFX5200 switches, and PTX1000 routers starting with Junos OS Release 17.2R1.

The Junos Telemetry Interface and gRPC streaming are supported on QFX5110, EX4600, and EX9200 switches starting with Junos OS Release 17.3R1.

Before you begin:

- Install Junos OS Release 16.1R3 or later on your Juniper Networks device.
- If your Juniper Networks device is running a version of Junos OS with an upgraded FreeBSD kernel, install the Junos Network Agent software package.
- Install the OpenConfig for Junos module. For more information see, *Installing the OpenConfig Package*.

To configure your system for gRPC services:

1. Specify the API connection setting either as unsecured or as based on Secure Socket Layer (SSL) technology. You can specify only one type of connection.

For example, to set the API connection as unsecured:

```
[edit system services]
user@host# set extension-service request-response grpc clear-text
```

For example, to set the API connection based on a SSL:

```
[edit system services]
user@host# set extension-service request-response grpc ssl
```

For an SSL-based connection, you must specify a local-certificate name. Optionally, you can specify an IP address to listen to for incoming connections (the default address is ::).

- a. Specify a local certificate-name, for example `jsd_certificate`:

```
[edit system services extension-service request-response grpc]
user@host# set ssl local-certificate jsd_certificate
```



NOTE: Enter the name of a certificate you have configured with the local *certificate-name* statement at the `[edit security certificates]` hierarchy level.

- b. (Optional) Specify an IP address to listen to for incoming connections. for example, `192.0.2.0`:

```
[edit system services extension-service request-response grpc]
user@host# set ssl ip-address 192.0.2.0
```



NOTE: If you do not specify an IP address, the default address of `::` is used to listen for incoming connections.

2. Specify port 50051 for accepting incoming connections through gRPC.



NOTE: Port 50051 is the required port for gRPC streaming for both unsecured and SSL-based connections.

```
[edit system services extension-service request-response grpc]
user@host# set ssl port 50051
```

3. Specify client IP addresses from which notifications are allowed, for example, `0.0.0.0`, which means any IP address:

```
[edit system services extension-service]
```

```
user@host# set notification allow-clients address 0.0.0.0
```

4. (Optional) Configure a WAN interface to your Juniper Networks device.



BEST PRACTICE: We recommend that you use a WAN interface to connect your Juniper Networks device to the management station you configure to collect telemetry data.

- See Also**
- [Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 58](#)
 - [Importing SSL Certificates for Junos XML Protocol Support](#)

Configuring Bidirectional Authentication for gRPC for Junos Telemetry Interface

Starting with Junos OS Release 17.4R1, you can configure bidirectional authentication for gRPC sessions used to stream telemetry data. Previously, only authentication of the server, that is, Juniper device, was supported. Now the external client, that is management station that collects data, can also be authenticated using SSL certificates. The JET service process (**jsd**), which supports application interaction with Junos OS, uses the credentials provided by the external client to authenticate the client and authorize a connection.

Before you begin:

- If your Juniper device is running a version of Junos OS with an upgraded FreeBSD kernel, install the Junos Network Agent software package.
- Install the OpenConfig for Junos module. For more information see, *Installing the OpenConfig Package*.
- Configure the gRPC server. For more information, see [“Configuring gRPC for the Junos Telemetry Interface” on page 64](#).

To configure authentication for the external client, that is, management station that collects telemetry data streamed from the Juniper device:

1. Enable bidirectional authentication and specify the requirements for a client certificate.

For example, to specify the strongest authentication, which requires a certificate and its validation:

```
[edit system services extension-service request-response grpc ssl]
user@host# set mutual-authentication client-certificate-request
require-certificate-and-verify
```




NOTE: The default is no-certificate. The other options are: request-certificate, request-certificate-and-verify, require-certificate, require-certificate-and-verify.

We recommend that you use no-certificate option in a test environment only.

2. Specify the certificate authority.



NOTE: For the certificate authority, specify a certificate-authority profile you have configured at the [edit security pki ca-profile] hierarchy level. This profile is used to validate the certificate provided by the client.

A digital certificate provides a way of authenticating users through a trusted third-party called a certificate authority (CA). The CA validates the identity of a certificate holder and “signs” the certificate to attest that it has not been forged or altered. For more information, see *Digital Certificates Overview* and *Example: Requesting a CA Digital Certificate*.

For example, to specify a certificate-authority profile named `jsd_certificate`:

```
[edit system services extension-service request-response grpc ssl
 mutual-authentication]
user@host# set certificate-authority jsd_certificate
```

3. Verify that an external client can successfully connect with the Juniper device through the `jsd` process and invoke OpenConfig RPCs.

The external client passes username and password credentials as part of metadata in each RPC. The RPC is allowed if valid credentials are used. Otherwise an error message is returned.

```
def getValidCredentialsMetadata(channel):
    metadata = [(b'username', b'user'), (b'password', b'pass')]

    stub = openconfig_service_pb2.beta_create_OpenconfigRpcApi_stub(channel)
    get_request = openconfig_service_pb2.GetRequestList(operation_id="1",
    operation=GET_CONFIG,

    path="/configuration/system")
    request = openconfig_service_pb2.GetRequest(request_id=1000, encoding=0,
    get_request=get_request)
    try:
        response = stub.Get(request, _TIMEOUT_SECONDS, metadata=metadata)
    except Exception as E:
        print E

    channel = grpc.insecure_channel(IP_ADDR + ":" + PORT)
    getValidCredentialsMetadata(channel)
```

See Also • [ssl on page 69](#)

ssl

```
Syntax  ssl {
        address ip-address;
        local-certificate local-certificate
        mutual-authentication {
            client-certificate-request {
                no-certificate;
                request-certificate;
                request-certificate-and-verify;
                require-certificate;
                require-certificate-and-verify;
            }
        }
        certificate-authority certificate-authority-profile-name;
        port port;
    }
```

Hierarchy Level [edit system services extension-service request-response grpc]

Release Information Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.
mutual-authentication, **client-certificate-request**, and **certificate-authority** options introduced in Junos OS Release 17.4R1.

Description Configure API connection settings based on Secure Sockets Layer (SSL) technology.

Options **address *ip-address***—Specify the IP address to listen for incoming connections. If you use the default IP address 0.0.0.0, the JET service process (jsd) listens on the IP address in the default routing instance.

Default: 0.0.0.0

mutual-authentication—Enable bidirectional authentication. Use this option, in conjunction with **client-certificate-request** and **certificate-authority *profile-name*** to configure client authentication using SSL-based certificates.

client-certificate-request—Specify the requirements for a client certificate.

no-certificate—Client certificate is not requested.



NOTE: We strongly recommend that you use this option in a test environment only.

request-certificate—Request certificate from client but do not verify.

request-certificate-and-verify—Request certificate from client and verify if provided.

require-certificate—Client certificate is mandatory, but do not verify.

require-certificate-and-verify—Client certificate is mandatory, and certificate is verified.

Default: no-certificate



NOTE: You can specify only one value for a client certificate.

certificate-authority *profile-name*—Specify the name of a certificate-authority profile configured at the [edit security pki ca-profile] hierarchy level. This profile is used to validate the certificate provided by the client.

port *port*—Specify the port number to accept incoming connections.



NOTE: For gRPC connections used to stream telemetry data, the required port number is 50051.

Range: 1 through 65535

Default: 9090

The remaining statement is explained separately. See [CLI Explorer](#).

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

Related Documentation	<ul style="list-style-type: none">• <i>grpc</i>• <i>JET Service Process Overview</i>• <i>Configuring Request-Response Service for JET Applications</i>
------------------------------	--

Guidelines for gRPC Sensors (Junos Telemetry Interface)

Starting with Junos OS Release 16.1R3, the Junos Telemetry Interface supports gRPC remote procedure calls (gRPC) to provision sensors and to subscribe to and receive telemetry data on MX Series routers and PTX3000 and PTX5000 routers.

Starting with Junos OS Release 17.2R1, QFX10000 switches, QFX5200 switches, and PTX1000 routers are also supported.

Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600 and EX9200 switches and the Routing and Control Board (RCB) on PTX3000 routers are also supported.

Starting with Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensors are supported.

Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are also supported.

See [Table 7 on page 71](#) for information about which sensors are supported with gRPC and on which platforms.

See [Table 8 on page 105](#) for a description of supported broadband edge (BBE) gRPC sensors, which are supported on all platforms supporting gRPC unless otherwise noted.

To activate a sensor, use the corresponding resource path. Each resource path enables data streaming for the system resource globally, that is, systemwide. You can also modify each resource path, such as to specify a specific logical or physical interface. For example, to specify a specific interface, include the following at the end of the path:

[name='interface-name']/

Supported gRPC Sensors

See [Table 7 on page 71](#) for a description of supported gRPC sensors and [Table 8 on page 105](#) for a description of supported broadband edge (BBE) gRPC sensors, including the subscription path you use to provision the sensors.

Table 7: gRPC Sensors

resource path	Description
<code>/junos/services/label-switched-path/usage/</code>	<p>Sensor for LSP statistics. On MX Series routers only, the following are also supported: bidirectional LSPs for ultimate-hop popping (UHP).</p> <p>Starting with Junos OS Release 17.2R1, QFX10000 switches and PTX1000 routers are also supported.</p> <p>Starting with Junos OS Release 17.3R1, EX9200 switches are also supported.</p> <p>Starting with Junos OS Release 17.4R1 on MX Series and PTX Series routers only, statistics for bypass LSPs are also exported. Previously, only statistics for ingress LSPs were exported.</p> <p>For bypass LSPs, the following are exported:</p> <ul style="list-style-type: none"> • Bypass LSP originating at the ingress router of the protected LSP. • Bypass LSP originating at the transit router of the protected LSP. • Bypass LSP protecting the transit LSP as well as the locally originated LSP. <p>When the bypass LSP is active, traffic is exported both on the bypass LSP and the ingress (protected) LSP.</p> <p>You can also specify an LSP name and source IP address at the end of the path: [name='lsp-name',source='ip-address']</p> <p>NOTE: When you enable a sensor for LSP statistics only, you must also configure the sensor-based-stats statement at the [edit protocols mpls] hierarchy level. MX Series routers should operate in enhanced mode. If not enabled by default, include either the enhanced-ip statement or the enhanced-ethernet statement at the [edit chassis network-services] hierarchy level.</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/network-instances/network-instance/mpls/</code>	<p>Sensor for LSP events and properties.</p> <p>Supported on MX Series and PTX Series routers and QFX10000 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on EX4600 and EX9200 switches and QFX5110 and QFX5200 switches starting with Junos OS Release 17.3R1.</p> <p>LSP events and properties are exported for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs.</p> <p>NOTE: Starting with Junos OS Release 17.4R1, telemetry data for LSP events and properties is reported separately for each routing instance. To export data for LSP events and properties, you must now include <code>/network-instances/network-instance/[name_'instance-name']/</code> in front of all supported paths. .</p> <p>The following paths are also supported:</p> <ul style="list-style-type: none"> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/lsp-instances/state/notify-status</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/state/notify-status</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/signaling-protocols/rsvp-te/sessions/session/state/notify-status</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/lsp-instances/state/bandwidth</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/lsp-instances/state/metric</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/state/explicit-path-name</code> <code>/network-instances/network-instance/[name_'instance-name']/mpls/lsp/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/lsp-instances/state/max-avg-bandwidth</code> <p>NOTE: To specify a specific LSP name and source address, include <code>[name='lsp-name',source='address']</code> after <code>mpls/lsp/constrained-path-tunnels/tunnel/</code> in any of the supported paths. If do not include a specific LSP name, data is exported for all configured LSPs.</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
/junos/npu-memory/	
junos/system/linecard/npu/memory/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for network processing unit (NPU) memory, NPU memory utilization, and total memory available for each memory type.</p> <p>Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1.</p> <p>Supported on EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>NOTE: Starting with Junos Release 17.4R1, FPC1 and FPC2 on PTX Series routers export data for NPU memory and NPU memory utilization. Previously, this sensor was supported only on FPC 3.</p> <p>The OpenConfig path is <code>/components/component[name="FPC<fpc-id>:NPU<npu-id>"]/properties/property/</code></p> <p>You can also add the following to the end of the path to stream specific statistics for NPU memory:</p> <ul style="list-style-type: none"> <code>[name="mem-util-<memory-name>-size"]/value</code> <code>[name="mem-util-<memory-name>-bytes-allocated"]/value</code> <code>[name="mem-util-<memory-name>-utilization"]/value</code> <code>[name="mem-util-<partition-name>-<app-name>-allocation-count"]/value</code> <code>[name="mem-util-<partition-name>-<app-name>-bytes-allocated"]/value</code> <code>[name="mem-util-<partition-name>-<app-name>-free-count"]/value</code> <p>You can add the following to the end of the path to stream specific statistics for NPU utilization:</p> <ul style="list-style-type: none"> <code>[name="util-<memory-name>-average-util"]>/value</code> <code>[name="util-<memory-name>-highest-util"]>/value</code> <code>[name="util-<memory-name>-lowest-util"]>/value</code> <code>[name="util-<memory-name>-average-cache-hit-rate"]>/value</code> <code>[name="util-<memory-name>-lowest-cache-hit-rate"]>/value</code> <code>[name="util-<packet-identifier>-rate"]>/value</code> <p>You can also export the following statistics for NPU memory for PTX routers only</p> <ul style="list-style-type: none"> <code>pfe_name</code> <code>combined_pool_name</code> <code>combined_size</code> <code>combined_usage_cnt</code> <code>combined_utilization</code> <code>global_pool_name</code> <code>global_usage_cnt</code> <code>global_alloc_cnt</code> <code>global_free_cnt</code> <code>local_pool_name</code> <code>local_usage_cnt</code> <code>local_alloc_cnt</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • <code>local_free_cnt</code>
<code>/junos/system/linecard/cpu/memory/</code>	<p>Sensor for CPU memory.</p> <p>NOTE: On PTX Series routers, FPC1 and FPC2 are not supported.</p> <p>Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1.</p> <p>Supported on EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>You can also include the following to end of the resource path for CPU memory:</p> <ul style="list-style-type: none"> • <code>[name="mem-util-<i>memory-name</i>-size"]/value</code> • <code>[name="mem-util-<i>memory-name</i>-bytes-allocated"]/value</code> • <code>[name="mem-util-<i>memory-name</i>-utilization"]/value</code> • <code>[name="mem-util-<i>memory-name</i>-<i>app-name</i>-allocations"]/value</code> • <code>[name="mem-util-<i>memory-name</i>-<i>app-name</i>-frees"]/value</code> • <code>[name="mem-util-<i>memory-name</i>-<i>app-name</i>-allocations-failed"]/value</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/network-instances/network-instance/protocols/protocol/ bgp/</code>	<p>NOTE: Starting with Junos OS Release 17.4R1 on MX Series and PTX Series routers, you can provision Junos Telemetry Interface sensors to export data for BGP routing tables (RIBs) for IPv4 and IPv6 routes.</p> <p>For BGP routing table paths, the <code>/network-instances/network-instance/</code> path is not supported.</p> <p>Each address family supports exporting data for five different tables, a main routing table, and four per-neighbor tables:</p> <ul style="list-style-type: none"> • <code>local-rib</code>—main BGP routing table for the main routing instance. • <code>adj-rib-in-pre</code>—NLRI updates received from the neighbor before any local input policy filters have been applied. • <code>adj-rib-in-post</code>—routes received from the neighbor eligible for best-path selection after local input policy filters have been applied. • <code>adj-rib-out-pre</code>—routes eligible for advertising to the neighbor before output policy filters have been applied. • <code>adj-rib-out-post</code>—routes eligible for advertising to the neighbor after output policy filters have been applied. <p>Use the following paths to export data for each BGP routing table. You can specify to export data either for IPv4 or IPv6 for each table:</p> <ul style="list-style-type: none"> • <code>/bgp-rib/afi-safis/afi-safi/ipv4-unicast/loc-rib/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv6-unicast/loc-rib/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv4-unicast/ neighbors/neighbor/adj-rib-in-pre/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv6-unicast/ neighbors/neighbor/adj-rib-in-pre/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv4-unicast/ neighbors/neighbor/adj-rib-in-post/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv6-unicast/ neighbors/neighbor/adj-rib-in-post/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv4-unicast/ neighbors/neighbor/adj-rib-out-pre/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv6-unicast/ neighbors/neighbor/adj-rib-out-pre/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv4-unicast/ neighbors/neighbor/adj-rib-out-post/</code> • <code>/bgp-rib/afi-safis/afi-safi/ipv6-unicast/ neighbors/neighbor/adj-rib-out-post/</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for BGP peer information.</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers, EX4600 and EX9200 switches, and QFX5110 switches starting with Junos OS Release 17.3R1.</p> <p>NOTE: Starting with Junos OS Release 17.3R1, telemetry data streamed through gRPC for BGP peers is reported separately for each configured routing instance.</p> <p>If your Juniper Network device is running Junos OS Release 17.3R1 or later, you must prepend the following to the beginning of any path you specify to stream statistics for BGP, with the exception of paths for routing tables: <code>/network-instances/network-instance[name_'instance-name']/protocols/protocol/</code></p> <p>Starting with Junos OS Release 17.3R1, the following paths are also supported:</p> <ul style="list-style-type: none"> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/prefixes/accepted</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/prefixes/rejected</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/active</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/queues/output</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/queues/input</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/snmp-peer-index</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/ImportEval</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/ImportEvalPending</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/messages/received/notification</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/messages/sent/notification</code> <code>/network-instances/network-instance/protocols/protocol/bgp/transport/state/remote-port</code> <code>/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/supported-capabilities</code> <p>NOTE: For all the following paths, with the exception of paths for routing tables, if your Juniper Networks device is running Junos OS Release 17.3R1 or later, you must prepend the following in front of the path: <code>/network-instances/network-instance[name_'instance-name']/protocols/protocol/</code></p> <p>You can also include the following at the end path to <code>/network-instances/network-instance[name_'instance-name']/protocols/protocol/bgp/neighbors/neighbor/</code>:</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> state/session-state state/messages/sent/update state/messages/received/update transport/state/local-address transport/state/remote-address state/peer-as afi-safis/afi-safi/state/prefix-limit/state/max-prefixes afi-safis/afi-safi/state/active state/session-status state/session-admin-status state/session-established-transitions state/interface-error state/prefix-limited-exceeded state/last-established established-transitions <p>You can also include the following at the end path to <code>/network-instances/network-instance[name_'instance-name']/protocols/protocol/bgp/global/</code>:</p> <ul style="list-style-type: none"> afi-safis/afi-safi/state/total-prefixes <p>You can also include the following at the end path to <code>/network-instances/network-instance[name_'instance-name']/protocols/protocol/bgp/peer-groups/peer-group[name_'peer-group-name']/</code>:</p> <ul style="list-style-type: none"> afi-safis/afi-safi/add-paths/eligible-prefix-policy state/peer-count/ <p>NOTE: For paths that export data for BGP routing tables, which are supported starting with Junos OS Release 17.4R1, you can append the following to each of the paths:</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • /num-routes • /routes/route/prefix • /routes/route/attributes • /routes/route/attributes/origin • /routes/route/attributes/as-path • /routes/route/attributes/next-hop • /routes/route/attributes/med • /routes/route/attributes/local-pref • /routes/route/attributes/atomic-aggr • /routes/route/attributes/aggregator/as • /routes/route/attributes/aggregator/as4 • /routes/route/attributes/aggregator/address • /routes/route/ext-attributes/ • /routes/route/ext-attributes/community • /routes/route/ext-attributes/originator-id • /routes/route/ext-attributes/cluster-list • /routes/route/ext-attributes/extended-community • /routes/route/ext-attributes/aigp • /routes/route/ext-attributes/path-id • /routes/route/ext-attributes/unknown-attribute • /routes/route/ext-attributes/unknown-attribute/attr-type • /routes/route/ext-attributes/unknown-attribute/attr-len • /routes/route/ext-attributes/unknown-attribute/attr-value • /routes/route/last-modified-date • /routes/route/last-update-received • /routes/route/valid-route • /routes/route/invalid-reason • /routes/route/best-path

Table 7: gRPC Sensors (*continued*)

resource path	Description
/junos/task-memory-information/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for memory utilization for routing protocol task.</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers, EX4600 and EX9200 switches and QFX5110 switches starting with Junos OS Release 17.3R1.</p> <p>You can also include the following at the end path to <code>/junos/task-memory-information/</code>:</p> <ul style="list-style-type: none"> task-memory-overall-report/task-size-block-list/task-size-block/tsb-size task-memory-overall-report/task-size-block-list/task-size-block/tsb-alloc-bytes task-memory-overall-report/task-size-block-list/task-size-block/tsb-allocs task-memory-overall-report/task-size-block-list/task-size-block/tsb-max-allocs task-memory-overall-report/task-size-block-list/task-size-block/tsb-max-bytes task-memory-overall-report/task-size-block-list/task-size-block/tsb-free-bytes task-memory-overall-report/task-memory-total-bytes task-memory-overall-report/task-memory-total-max-bytes task-memory-information/task-memory-overall-report/task-memory-total-free-bytes task-memory-allocator-report/task-block-list/task-block/tb-name task-memory-allocator-report/task-block-list/task-block/tb-size task-memory-allocator-report/task-block-list/task-block/tb-alloc-size task-memory-allocator-report/task-block-list/task-block/tb-alloc-blocks task-memory-allocator-report/task-block-list/task-block/tb-alloc-bytes task-memory-allocator-report/task-block-list/task-block/tb-max-alloc-blocks task-memory-allocator-report/task-lite-page-list/task-lite-page/tlp-name task-memory-allocator-report/task-lite-page-list/task-lite-page/tlp-alloc-bytes task-memory-allocator-report/task-memory-total-bytes task-memory-information/task-memory-allocator-report/task-memory-total-max-bytes task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-name task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-allocs task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-alloc-bytes task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-max-allocs task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-max-alloc-bytes task-memory-malloc-usage-report/task-malloc-list/task-malloc/tm-function-calls task-memory-malloc-usage-report/task-memory-total-bytes task-memory-malloc-usage-report/task-memory-total-max-bytes task-memory-max-dynamic-allocs task-memory-bss-bytes task-memory-max-bss-bytes task-memory-page-data-bytes task-memory-max-page-data-bytes task-memory-dir-bytes task-memory-max-dir-bytes task-memory-total-bytes-in-use

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> task-memory-total-bytes-percent
/junos/system/linecard/firewall/	<p>Sensor for firewall filter counters and policer counters. Each line card reports counters separately.</p> <p>Supported on QFX10000 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>NOTE: Hierarchical policer statistics are collected for MX Series routers only. Traffic-class counter statistics are collected for PTX Series routers and QFX10000 switches only.</p> <p>Firewall counters are exported even if the interface to which the firewall filter is attached is operationally down.</p> <p>The following OpenConfig paths are supported:</p> <ul style="list-style-type: none"> junos/firewall/firewall-stats/[name='filter-name']/timestamp /junos/firewall/firewall-stats/[name='filter-name']/memory-usage/[name='memory-type']/allocated /junos/firewall/firewall-stats/[name='filter-name']/counter-stats/[name='counter-name']/packets /junos/firewall/firewall-stats/[name='filter-name']/counter-stats/[name='counter-name']/bytes /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/out-of-spec-packets /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/out-of-spec-bytes /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/offered-packets /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/offered-bytes /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/transmitted-packets /junos/firewall/firewall-stats/[name='filter-name']/policer-stats/[name='policer-name']/transmitted-bytes /junos/firewall/firewall-stats/[name='filter-name']/hierarchical-policer-stats/[name='hierarchical-policer-name']/premium-packets (MX Series only) /junos/firewall/firewall-stats/[name='filter-name']/hierarchical-policer-stats/[name='hierarchical-policer-name']/premium-bytes (MX Series only) /junos/firewall/firewall-stats/[name='filter-name']/hierarchical-policer-stats/[name='hierarchical-policer-name']/aggregate-packets (MX Series only) /junos/firewall/firewall-stats/[name='filter-name']/hierarchical-policer-stats/[name='hierarchical-policer-name']/aggregate-bytes (MX Series only)

Table 7: gRPC Sensors (*continued*)

resource path	Description
/interfaces/interface/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for physical interface traffic.</p> <p>NOTE: For PTX Series routers, for a specific interface, queue statistics are exported for each line card. For MX series routers, interface queue statistics are exported only from slot on which an interface is configured.</p> <p>For Aggregated Ethernet interfaces, statistics are exported for the member physical interfaces. You must aggregate the counters at the destination server, or collector.</p> <p>If a physical interface is administratively down or operationally down, interface counters are not exported.</p> <p>Only fields with a non-zero value are exported.</p> <p>Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1.</p> <p>Supported on EX9200 switches and MX150 routers starting with Junos OS Release 17.3R1.</p> <p>The following paths are also supported:</p> <ul style="list-style-type: none"> • /interfaces/interface[name='interface-name']/parent_ae_name • /interfaces/interface[name='interface-name']/oper-status • /interfaces/interface[name='interface-name']/carrier-transitions • /interfaces/interface[name='interface-name']/last-change • /interfaces/interface[name='interface-name']/high-speed • /interfaces/interface[name='interface-name']/counters/out-octets • /interfaces/interface[name='interface-name']/counters/out-unicast-pkts • /interfaces/interface[name='interface-name']/counters/out-multicast-pkts • /interfaces/interface[name='interface-name']/counters/out-broadcast-pkts • /interfaces/interface[name='interface-name']/counters/out-errors • /interfaces/interface[name='interface-name']/counters/in-octets • /interfaces/interface[name='interface-name']/counters/in-unicast-pkts • /interfaces/interface[name='interface-name']/counters/in-multicast-pkts • /interfaces/interface[name='interface-name']/ • /interfaces/interface[name='interface-name']/counters/in-broadcast-pkts • /interfaces/interface[name='interface-name']/counters/in-errors • /interfaces/interface[name='interface-name']/in-pause-pkts • /interfaces/interface[name='interface-name']/out-pause-pkts • /interfaces/interface[name='interface-name']/in-queue [queue-number=queue_number]/ • /interfaces/interface[name='interface-name']/in-queue [queue-number=queue_number]/ pkts • /interfaces/interface[name='interface-name']/in-queue [queue-number=queue_number]/bytes • /interfaces/interface[name='interface-name']/in-queue [queue-number=queue_number]/tail-drop-pkts • /interfaces/interface[name='interface-name']/in-queue

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> [queue-number=<i>quene_number</i>]/ rl-drop-pkts • /interfaces/interface[name='interface-name']/in-queue [queue-number=<i>quene_number</i>]/ rl-drop-bytes • /interfaces/interface[name='interface-name']/in-queue [queue-number=<i>quene_number</i>]/avg-buffer-occupancy • /interfaces/interface[name='interface-name']/in-queue [queue-number=<i>quene_number</i>]/cur-buffer-occupancy • /interfaces/interface[name='interface-name']/in-queue [queue-number=<i>quene_number</i>]/peak-buffer-occupancy • /interfaces/interface[name='interface-name']/in-queue [queue-number=<i>quene_number</i>]/allocated-buffer-size • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/pkts • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/bytes • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/tail-drop-pkts • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/rl-drop-pkts • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/ rl-drop-bytes • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/red-drop-pkts • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/red-drop-bytes • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/avg-buffer-occupancy • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/cur-buffer-occupancy • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/ peak-buffer-occupancy • /interfaces/interface[name='interface-name']/out-queue [queue-number=<i>quene_number</i>]/allocated-buffer-size

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/interfaces/interface/subinterfaces/</code>	Sensor for logical interface traffic.
<code>/interfaces/interface[name='interface-name']/subinterfaces/</code>	<p>NOTE: If a logical interface is operationally down, interface statistics continue to be exported.</p> <p>NOTE: Locally injected packets from the Routing Engine are not exported.</p> <p>Supported on QFX10000 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>The following paths are also supported:</p> <ul style="list-style-type: none"> <code>/interfaces/interface/subinterfaces/subinterface/name</code> <code>/interfaces/interface/subinterfaces/subinterface/ifindex</code> <code>/interfaces/interface/subinterfaces/subinterface/snmp_index</code> <code>/interfaces/interface/subinterfaces/subinterface/admin_status</code> <code>/interfaces/interface/subinterfaces/subinterface/oper_status</code> <code>/interfaces/interface/subinterfaces/subinterface/last_change</code> <code>/interfaces/interface/subinterfaces/subinterface/high_speed</code> <code>/interfaces/interface/subinterfaces/subinterface/description</code> <code>/interfaces/interface/subinterfaces/subinterface/enabled</code> <code>/interfaces/interface/subinterfaces/subinterface/subunit</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_octets</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_unicast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_broadcast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_multicast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_discards</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_errors</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/in_unknown_protos</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_octets</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_unicast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_broadcast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_multicast_pkts</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_discards</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/out_errors</code> <code>/interfaces/interface/subinterfaces/subinterface/oob_states/last_clear</code>
<code>/junos/system/linecard/optics/</code>	<p>Sensor for various optical interface performance metrics, such as transmit and receive power levels.</p> <p>Supported on QFX10000 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1.</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/junos/rsvp-interface-information/</code>	<p>Sensor for events and properties for RSVP interfaces.</p> <p>NOTE: For 100 RSVP logical interfaces, configure a sampling interval equal to 60 seconds. For 200 RSVP logical interfaces, configure a sampling interval equal to 180 seconds.</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers, QFX5110 switches, and EX4600 and EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to the end path for <code>/junos/rsvp-interface-information/</code>:</p> <ul style="list-style-type: none"> • <code>active-count</code> • <code>rsvp-interface/interface-name</code> • <code>rsvp-interface/index</code> • <code>rsvp-interface/rsvp-status</code> • <code>rsvp-interface/authentication-flag</code> • <code>rsvp-interface/aggregate-flag</code> • <code>rsvp-interface/ack-flag</code> • <code>rsvp-interface/protect-flag</code> • <code>rsvp-interface/hello-interval</code> • <code>rsvp-interface/interface-address</code> • <code>message-statistics/rsvp-message</code> • <code>rsvp-interface/message-statistics/messages-sent</code> • <code>rsvp-interface/message-statistics/messages-received</code> • <code>rsvp-interface/message-statistics/messages-sent-5seconds</code> • <code>rsvp-interface/message-statistics/messages-received-5seconds</code> • <code>rsvp-interface/rsvp-telink/active-reservation</code> • <code>rsvp-interface/rsvp-telink/preemption-count</code> • <code>rsvp-interface/rsvp-telink/update-threshold</code> • <code>rsvp-interface/rsvp-telink/subscription</code> • <code>rsvp-interface/rsvp-telink/static-bandwidth</code> • <code>rsvp-interface/rsvp-telink/available-bandwidth</code> • <code>rsvp-interface/rsvp-telink/reserved-bandwidth/bandwidth-priority</code> • <code>rsvp-interface/rsvp-telink/reserved-bandwidth/total-reserved-bandwidth</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
/components/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for operational state of Routing Engines, power supply modules, Switch Fabric Boards, Control Boards, Switch Interface Boards, Modular Interface Cards, and Physical Interface Cards.</p> <p>NOTE:</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on EX9200 switches and MX150 routers starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to each of the paths:</p> <ul style="list-style-type: none"> • name • cidr • version • part_number • serial_number • description • clei_code • model • vendor_name • properties/property/state • properties/property/state_offline_reason (MX Series only) • properties/property/power_usage • properties/property/power_maximum • properties/property/temperature_intake • properties/property/temperature_exhaust_a (not supported on PTX1000 and PTX3000 routers) • properties/property/temperature_exhaust_b (not supported on PTX1000 and PTX3000 routers) • properties/property/temperature_exhaust (not supported on PTX1000 and PTX5000 routers) • properties/property/cpu_utilization_total • properties/property/memory_dram_used • properties/property/memory_utilization_heap • properties/property/memory_utilization_buffer • properties/property/uptime <p>The following paths are also supported only for Routing Engine statistics:</p> <ul style="list-style-type: none"> • properties/property/mastership-state • properties/property/mastership-priority • properties/property/temperature-cpu • properties/property/memory-dram-installed • properties/property/cpu-utilization-user • properties/property/cpu-utilization-background • properties/property/cpu-utilization-kernel

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • <code>properties/property/cpu-utilization-idle</code> • <code>properties/property/reboot-reason</code> <p>The following paths are also supported for power modules:</p> <ul style="list-style-type: none"> • <code>properties/property/power-zone-upper-capacity</code> • <code>properties/property/power-zone-upper-maximum</code> • <code>properties/property/power-zone-upper-allocated</code> • <code>properties/property/power-zone-upper-remaining</code> • <code>properties/property/power-zone-upper-usage</code> • <code>properties/property/power-zone-lower-capacity</code> • <code>properties/property/power-zone-lower-maximum</code> • <code>properties/property/power-zone-lower-allocated</code> • <code>properties/property/power-zone-lower-remaining</code> • <code>properties/property/power-zone-lower-usage</code> • <code>properties/property/power-zone-0-capacity</code> • <code>properties/property/power-zone-0-maximum</code> • <code>properties/property/power-zone-0-allocated</code> • <code>properties/property/power-zone-0-remaining</code> • <code>properties/property/power-zone-0-usage</code> • <code>properties/property/power-zone-1-capacity</code> • <code>properties/property/power-zone-1-maximum</code> • <code>properties/property/power-zone-1-allocated</code> • <code>properties/property/power-zone-1-remaining</code> • <code>properties/property/power-zone-1-usage</code> • <code>properties/property/power-system-capacity</code> • <code>properties/property/power-system-allocated</code> • <code>properties/property/power-system-remaining</code> • <code>properties/property/power-system-usage</code> • <code>properties/property/temperature-ambient</code> <p>The following paths are supported for either Switch Fabric Board or Control Boards or both:</p> <ul style="list-style-type: none"> • <code>properties/property/temperature-zone-0-intake</code> (SFB only) • <code>properties/property/temperature-zone-0-intake-a</code> (both SFB and CB) • <code>properties/property/temperature-zone-1-intake-b</code> (both SFB and CB) • <code>properties/property/temperature-zone-0-exhaust</code> (SFB only) • <code>properties/property/temperature-zone-1-exhaust</code> (SFB only) • <code>properties/property/temperature-zone-0-intake-c</code> (CB only) • <code>properties/property/temperature-zone-0-exhaust-a</code> (CB only) • <code>properties/property/temperature-zone-1-exhaust-b</code> (CB only)

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/lacp/</code>	<p>Sensor for operational state of aggregated Ethernet interfaces configured with the Link Aggregation Control Protocol.</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to the end of the path for <code>/lacp/</code>:</p> <ul style="list-style-type: none"> • <code>state/system-priority</code> • <code>interfaces/interface[name='aggregate-interface-name']/state/</code> • <code>interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/</code> • <code>interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/counters/</code> • <code>interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/port-num</code> • <code>interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/partner-port-num</code> • <code>interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/mux-state</code>
<code>/lldp/</code>	<p>Sensor for operational state of Ethernet interfaces enabled with the Link Layer Discovery Protocol.</p> <p>Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers and EX9200, EX4600, and QFX5110 switches starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to the end of the path for <code>/lldp/</code>:</p> <ul style="list-style-type: none"> • <code>state/</code> • <code>state/enabled</code> • <code>state/hello-timer</code> • <code>state/system-name</code> • <code>state/system-description</code> • <code>state/chassis-id</code> • <code>state/loc-port-id-type</code> • <code>interfaces/interface[name='interface-name']/state/</code> • <code>interfaces/interface[name='interface-name']/state/counters/</code> • <code>interfaces/interface[name='interface-name']/neighbors/</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/arp-information/</code>	<p>Sensor for Address Resolution Protocol (ARP) statistics.</p> <p>Supported on QFX10000 and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers, EX9200 switches, and MX150 routers starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to the end path for <code>/arp-information/</code></p> <ul style="list-style-type: none"> • <code>ipv4</code> • <code>ipv4/neighbors</code> • <code>ipv4/neighbors/neighbor</code> • <code>ipv4/neighbors/neighbor/ip</code> • <code>ipv4/neighbors/neighbor/link-layer-address</code> • <code>pv4/neighbors/neighbor/origin</code> • <code>ipv4/neighbors/neighbor/host-name</code> • <code>ipv4/neighbors/neighbor/rtr-id</code> • <code>ipv4/neighbors/neighbor/state</code> • <code>ipv4/neighbors/neighbor/expiry</code> • <code>ipv4/neighbors/neighbor/ispublish</code> • <code>ipv4/neighbors/neighbor/interface-name</code> • <code>ipv4/neighbors/neighbor/logical-router-id</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
/interfaces/interface[name='interface-name']/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for Routing Engine internal interfaces.</p> <p>NOTE: On MX Series routers, you can specify the following interfaces: fxp0, em0, and em1</p> <p>On PTX Series routers, you can specify the following interfaces: em0, ixlv0, ixlv1</p> <p>On PTX Series routers with dual Routing Engines, you can specify the following interfaces: em0, ixgbe0, ixgbe1</p> <p>Support on PTX1000 routers starting with Junos OS Release 17.3R1.</p> <p>The following end paths are also supported:</p> <ul style="list-style-type: none"> • interfaces/interface/state/type • /interfaces/interface/state/mtu • /interfaces/interface/state/name • /interfaces/interface/state/description • /interfaces/interface/state/enabled • /interfaces/interface/state/ifindex • /interfaces/interface/state/admin-status • /interfaces/interface/state/oper-status • /interfaces/interface/state/last-change • /interfaces/interface/state/speed • /interfaces/interface/state/counters/in-octets • /interfaces/interface/state/counters/in-unicast-pkts • /interfaces/interface/state/counters/in-broadcast-pkts • /interfaces/interface/state/counters/in-multicast-pkts • /interfaces/interface/state/counters/in-discards • /interfaces/interface/state/counters/in-errors • /interfaces/interface/state/counters/in-unknown-protos • /interfaces/interface/state/counters/out-octets • /interfaces/interface/state/counters/out-unicast-pkts • /interfaces/interface/state/counters/out-broadcast-pkts • /interfaces/interface/state/counters/out-multicast-pkts • /interfaces/interface/state/counters/out-discards • /interfaces/interface/state/counters/out-errors • /interfaces/interface/state/counters/last-clear • /interfaces/interface/state/counters/in-pkts • /interfaces/interface/state/counters/in-sec-pkts • /interfaces/interface/state/counters/in-sec-octets • /interfaces/interface/state/counters/in-pause-pkts • /interfaces/interface/state/counters/out-pkts • /interfaces/interface/state/counters/out-sec-pkts • /interfaces/interface/state/counters/out-sec-octets • /interfaces/interface/state/counters/out-pause-pkts

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • /interfaces/interface/state/counters/in-drops • /interfaces/interface/state/counters/in-frame-errors • /interfaces/interface/state/counters/in-runs • /interfaces/interface/state/counters/in-lchan-errors • /interfaces/interface/state/counters/in-l-mismatch-errors • /interfaces/interface/state/counters/in-fifo-errors • /interfaces/interface/state/counters/in-giants • /interfaces/interface/state/counters/in-resource-errors • /interfaces/interface/state/counters/out-drops • /interfaces/interface/state/counters/carrier-transitions • /interfaces/interface/state/counters/mtu-errors • /interfaces/interface/state/counters/out-resource-errors • /interfaces/interface/subinterfaces/subinterface/index • /interfaces/interface/subinterfaces/subinterface/state/index • /interfaces/interface/subinterfaces/subinterface/state/name • /interfaces/interface/subinterfaces/subinterface/state/description • /interfaces/interface/subinterfaces/subinterface/state/enabled • /interfaces/interface/subinterfaces/subinterface/state/ifindex • /interfaces/interface/subinterfaces/subinterface/state/admin-status • /interfaces/interface/subinterfaces/subinterface/state/oper-status • /interfaces/interface/subinterfaces/subinterface/state/last-change • /interfaces/interface/subinterfaces/subinterface/state/counters/in-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/in-octets • /interfaces/interface/subinterfaces/subinterface/state/counters/in-unicast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/in-broadcast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/in-multicast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/in-discards • /interfaces/interface/subinterfaces/subinterface/state/counters/in-errors • /interfaces/interface/subinterfaces/subinterface/state/counters/in-unknown-protos • /interfaces/interface/subinterfaces/subinterface/state/counters/out-octets • /interfaces/interface/subinterfaces/subinterface/state/counters/out-unicast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/out-broadcast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/out-multicast-pkts • /interfaces/interface/subinterfaces/subinterface/state/counters/out-discards • /interfaces/interface/subinterfaces/subinterface/state/counters/out-errors • /interfaces/interface/subinterfaces/subinterface/state/counters/last-clear • /interfaces/interface/subinterfaces/subinterface/state/counters/out-pkts

Table 7: gRPC Sensors (*continued*)

resource path	Description
<code>/nd6-information/</code>	<p>Sensor for Network Discovery Protocol (NDP) table state.</p> <p>Supported on QFX10000 and QFX5200 switches starting with Junos OS Release 17.2R1.</p> <p>Supported on PTX1000 routers, EX9200 switches, and MX150 routers starting with Junos OS Release 17.3R1.</p> <p>You can also add the following to the end path for nd6-information/</p> <ul style="list-style-type: none"> • <code>ipv6</code> • <code>ipv6/neighbors</code> • <code>ipv6/neighbors/neighbor</code> • <code>ipv6/neighbors/neighbor/ip</code> • <code>ipv6/neighbors/neighbor/link-layer-address</code> • <code>ipv6/neighbors/neighbor/origin</code> • <code>ipv6/neighbors/neighbor/isrouter</code> • <code>ipv6/neighbors/neighbor/state</code> • <code>ipv6/neighbors/neighbor/rtb-id</code> • <code>ipv6/neighbors/neighbor/issecure</code> • <code>ipv6/neighbors/neighbor/ispublish</code> • <code>ipv6/neighbors/neighbor/expiry</code> • <code>ipv6/neighbors/neighbor/interface-name</code> • <code>ipv6/neighbors/neighbor/logical-router-id</code>
<code>/ipv6-ra/</code>	Sensor for NDP router-advertisement statistics.
<code>/junos/system/linecard/packet/usage/</code>	<p>Sensor for Packet Forwarding Engine Statistics. This sensor exports statistics for counters and provides visibility into Packet Forwarding Engine error and drop statistics.</p> <p>This sensor is supported starting on MX Series and PTX Series routers starting with Junos OS Release 17.4R1.</p>

Table 7: gRPC Sensors (*continued*)

resource path	Description
/network-instances/network-instance/protocols/protocol/ isis/levels/level/	
/network-instances/network-instance/protocols/protocol/ isis/interfaces/interface/levels/level/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for IS-IS routing protocol statistics. Statistics are exported separately for each routing instance.</p> <p>To specify a routing-instance name:</p> <p><code>/network-instances/network-instance[name_ 'instance-name']/protocols/protocol/isis/levels/level/</code></p> <p><code>/network-instances/network-instance[name_ 'instance-name']/protocols/protocol/isis/interfaces/interface/levels/level/</code></p> <p>NOTE: This sensor is supported on MX Series and PTX Series routers starting with Junos OS Release 17.4R1.</p> <p>The following paths are also supported:</p> <ul style="list-style-type: none"> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/received</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/processed</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/dropped</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/sent</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/retransmit</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/received</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/processed</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/dropped</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/sent</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/retransmit</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/received</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/processed</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/dropped</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/sent</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/retransmit</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/cnsp/received</code> <code>/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/cnsp/processed</code> <code>/network-instances/network-instance/protocols/protocol/</code>

Table 7: gRPC Sensors (*continued*)

resource path	Description
	isis/interfaces/interface/levels/level/packet-counters/cnsp/dropped
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/cnsp/sent
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/corrupted-lsps
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/database-overloads
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/manual-address-drop-from-area
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/exceeded-max-seq-nums
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/seq-num-skips
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/own-lsp-purges
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/id-len-mismatch
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/part-changes
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/max-area-address-mismatches
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/auth-fails
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/spf-runs
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/auth-type-fails
	• /network-instances/network-instance/protocols/protocol/isis/levels/level/system-level-counters/state/lsp-errors
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/adj-changes
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/adj-number
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/auth-fails
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/auth-type-fails
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/id-field-len-mismatches
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/lan-dis-changes
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/max-area-address-mismatch
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/circuit-counters/state/rejected-adj
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/system-id
	• /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/dis-system-id

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/local-extended-system-id • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/neighbor-extended-system-id • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/adjacency-state • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/neighbor-circuit-type • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/neighbor-ipv4-address • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/neighbor-ipv6-address • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/neighbor-snpa • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/levels/level/adjacencies/adjacency/state/priority • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/remaining-hold-time • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/restart-status • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/restart-support • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/restart-suppress • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/up-time • /network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/adjacencies/adjacency/state/nlpid • /network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/adjacencies/adjacency/state/area-address • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/topologies • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/multi-topology • /network-instances/network-instance/protocols/protocol/isis/interfaces/interfaces/levels/level/adjacencies/adjacency/state/adjacency-type • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/state/ipv4-prefix • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/state/up-down • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/state/s-bit • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/state/metric

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/flags • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/ipv4-source-router-id • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/ipv6-source-router-id/state/ipv6-source-router-id • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/tag64 • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/tag32 • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/type • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/length • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/value • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/prefix-sid/sid/state/value • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/flags • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/extended-ipv4-reachability/prefixes/prefix/subtlvs/subtlv/flags/state/algorithm • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/ipv6-prefix • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/up-down • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/s-bit • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/x-bit • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/metric

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/state/ipv6-prefix • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/ipv4-source-router-id/state/ipv4-source-router-id • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/ipv6-source-router-id/state/ipv6-source-router-id • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/tag64/state/tag64 • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/tag64/state/tag32 • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/type • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/length • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/undefined-subtlvs/undefined-subtlv/state/value • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/prefix-sid/sid/state/value • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/prefix-sid/sid/state/flags • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/ipv6-reachability/prefixes/prefix/subtlvs/subtlv/prefix-sid/sid/state/algorithm • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/router-capabilities/router-capability/state/flags • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/tlvs/tlv/router-capabilities/router-capability/state/rtr-id • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/undefined-tlvs/undefined-tlv/state/type • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/undefined-tlvs/undefined-tlv/state/length • /network-instances/network-instance/protocols/protocol/isis/levels/level/link-state-database/lsp/undefined-tlvs/undefined-tlv/state/value

Table 7: gRPC Sensors (*continued*)

resource path	Description
/junos/services/segment-routing/interface/ingress/usage/	
/junos/services/segment-routing/interface/egress/usage/	
/junos/services/segment-routing/sid/usage/	

Table 7: gRPC Sensors (*continued*)

resource path	Description
	<p>Sensors for aggregate segment routing traffic with IS-IS.</p> <p>This sensor is supported on MX Series and PTX5000 routers starting with Junos OS Release 17.4R1.</p> <p>Statistics are exported separately for each routing instance.</p> <p>The first path exports inbound traffic. The second path exports outbound traffic. The third path exports inbound segment routing traffic for each segment identifier.</p> <p>NOTE: When you enable a sensor for segment routing statistics, you must also configure the sensor-based-stats statement at the [edit protocols isis source-packet-routing] hierarchy level. MX Series and PTX Series routers must also operate in enhanced mode. On MX Series routers, if not enabled by default, configure either the enhanced-ip statement or the enhanced-ethernet statement at the [edit chassis network-services] hierarchy level. On PTX Series routers, configure the enhanced-mode statement at the [edit chassis network-services] hierarchy level.</p> <p>NOTE: Currently, MPLS labels correspond only to only one instance, instance 0. Since each SID corresponds to a single instance_identifier, no aggregation is required to be done by the collector. The instance_identifier is stamped as 0.</p> <p>The following OpenConfig paths are supported:</p> <ul style="list-style-type: none"> /network-instances/network-instance/mpls/signaling-protocols/segment-routing/interfaces/interface/state/in-pkts /network-instances/network-instance/mpls/signaling-protocols/segment-routing/interfaces/interface/state/in-octets /network-instances/network-instance/mpls/signaling-protocols/segment-routing/interfaces/interface/state/out-octets /network-instances/network-instance/mpls/signaling-protocols/segment-routing/interfaces/interface/state/out-pkts /network-instances/network-instance/mpls/aggregate-sid-counters/aggregate-sid-counter/state/in-octets /network-instances/network-instance/mpls/aggregate-sid-counters/aggregate-sid-counter/state/in-pkts /network-instances/network-instance/mpls/aggregate-sid-counters/aggregate-sid-counter/state/out-octets /network-instances/network-instance/mpls/aggregate-sid-counters/aggregate-sid-counter/state/out-pkts /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/state/in-octets /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/state/in-pkts /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/state/out-octets /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/state/out-pkts /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/forwarding-classes/forwarding-class/state/

Table 7: gRPC Sensors (*continued*)

resource path	Description
	in-octets <ul style="list-style-type: none"> • /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/forwarding-classes/forwarding-class/state/in-pkts • /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/forwarding-classes/forwarding-class/state/out-octets • /network-instances/network-instance/mpls/interfaces/interface/sid-counters/sid-counter/forwarding-classes/forwarding-class/state/out-pkts

Table 8: Broadband Edge gRPC Sensors

resource path	Description
/junos/system/subscriber-management/aaa/accounting-statistics/	Sensor that tracks accounting statistics by means of a protocol exchange with accounting servers. <p>You can also add the following to the end path for /junos/system/subscriber-management/aaa/accounting-statistics/:</p> <ul style="list-style-type: none"> • acct-req-received • acct-req-timeout • acct-resp-failure • acct-resp-success

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/aaa/ address-assignment-statistics/ logical-system-routing-instances/ logical-system-routing-instance/pools/pool</code>	<p>For Authentication, Authorization, and Accounting, this sensor tracks address pool utilization.</p> <p>The resource path can be refined to select a logical system routing instance by using a logical system routing instance filter:</p> <p><code>/aaa/address-assignment-statistics/logical-system-routing-instances/ logical-system-routing-instance [lsri-name='lsName:riName']/pools/ pool[pool-name='poolName']</code></p> <p>The resource path can be refined to select a specific pool by using a pool filter:</p> <p><code>/junos/system/subscriber-management/aaa/address-assignment-statistics/ logical-system-routing-instances/logical-system-routing-instance/pools/ pool[pool-name='poolName']</code></p> <p>The resource path can be refined to select both a logical routing instance and a pool by using a logical system routing instance filter and a pool filter:</p> <p><code>/junos/system/subscriber-management/aaa/address-assignment-statistics/ logical-system-routing-instances/logical-system-routing-instance/ [lsri-name='lsName:riName']/pools/pool[pool-name='poolName']</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • pool-name • out-of-memory • out-of-address • address-total • address-in-use • address-usage-percent

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/aaa/authentication-statistics/</code>	<p>Sensors that track authentication, authorization, and accounting (AAA) authentication, pre-authentication, and re-authentication statistics.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • req-received • req-accepted • req-rejected • req-challenge • req-timeout • pre-authen-req-received • pre-authen-req-accepted • pre-authen-req-rejected • pre-authen-req-challenge • pre-authen-req-timeout • re-authen-req-received • re-authen-req-accepted • re-authen-req-rejected • re-authen-req-internal-errors • re-authen-req-challenge • re-authen-req_timeout
<code>/junos/system/subscriber-management/aaa/dynamic-request-statistics/</code>	<p>Sensor tracks dynamic request statistics from AAA server-initiated requests, including Change of Authorization (CoA) and RADIUS-initiated Disconnect (RID).</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dynamic-req-received • dynamic-req-success • dynamic-req-error • dynamic-req-silently-drop

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/aaa/radius-servers/radius-server/response-time/</code>	<p>Sensor for RADIUS server response time statistics for a specific server.</p> <p>A request sent to the RADIUS server is counted as a message sent. Similarly, a response to the request is counted as a message received. A timeout during the measurement interval does not impact the minimum, average, or maximum response time statistics, but the event is counted as a no response.</p> <p>The delay measurements are made over a 60-second measurement interval. The reporting interval can be as much as 59 seconds out of phase with the measurement interval. At reporting time, the values from the last update interval are reported. The response time values are not aligned with the reporting interval.</p> <p>The resource path can be refined to select a specific RADIUS server by adding a server address filter to the resource path:</p> <p><code>/junos/system/subscriber-management/aaa/radius-servers/radius-server[server-address='radius/pv4Address']/response-time/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>one-minute-minimum-response-time</code> • <code>one-minute-average-response-time</code> • <code>one-minute-maximum-response-time</code> • <code>one-minute-messages-sent</code> • <code>one-minute-messages-received</code> • <code>one-minute-messages-no-response</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/aaa/ radius-servers/radius-server/statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for RADIUS server statistics for a specific server.</p> <p>The resource path can be refined to select a specific RADIUS server by adding a server address filter to the resource path:</p> <pre>/junos/system/subscriber-management/aaa/radius-servers/ radius-server[server-address='radius/pv4Address']/statistics/</pre> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • server-address • server-last-rtt • auth-access-requests • auth-rollover-requests • auth-retransmissions • auth-access-accepts • auth-access-rejects • auth-access-challenges • auth-malformed-responses • auth-bad-authenticators • auth-req-pending • auth-request-timeouts • auth-unknown-responses • auth-packets-dropped • preauth-access-requests • preauth-rollover-requests • preauth-retransmissions • preauth-access-accepts • preauth-access-rejects • preauth-access-challenges • preauth-malformed-responses • preauth-bad-authenticators • preauth-req-pending • preauth-request-timeouts • preauth-unknown-responses • preauth-packets-dropped • acct-start-requests • acct-interim-requests • acct-stop-requests • acct-rollover-requests • acct-retransmissions • acct-start-responses • acct-interim-responses • acct-stop-responses • acct-malformed-responses • acct-bad-authenticators

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • acct-req-pending • acct-request-timeouts • acct-unknown-responses • acct-packets-dropped
/junos/system/subscriber-management/client-protocols/ dhcp/v4/routing-instances/routing-instance/relay/ bindings/	<p>Sensor for DHCPv4 relay binding state statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance name:</p> <p>/junos/system/subscriber-management/client-protocols/dhcp/v4/ routing-instances/routing-instance[name='routing-instance-name']/relay/ bindings/</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • binding-state-v4relay-binding • binding-state-v4relay-init • binding-state-v4relay-bound • binding-state-v4relay-selecting • binding-state-v4relay-requesting • binding-state-v4relay-renew • binding-state-v4relay-release • binding-state-v4relay-restoring

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/relay/servers/server/response-time</code>	<p>Sensor for DHVPv4 server delay. The sensor periodically measures the minimum, average, and maximum delay or response time from the upstream DHCP server(s), as seen by the relay.</p> <p>DHCP relay does not track the state of the server. The no-response statistics are the difference between the messages sent and received during the measurement interval.</p> <p>The delay measurements are made over a 60-second measurement interval. Because the reporting interval can be as much as 59 seconds out of phase with the measurement interval, there is no design to align the response time values with the reporting interval.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name='routing-instance-name']/relay/servers/server/response-time</code></p> <p>The resource path can be refined to select a specific DHCP server by adding a server filter to the resource path:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/relay/servers/server[server-ip='server-ip']/response-time</code></p> <p>The resource path can be refined to select a specific DHCP server in a specific routing instance by adding both a routing instance filter and a server filter to the resource path:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name='routing-instance-name']/relay/servers/server[server-ip='server-ip']/response-time</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>one-minute-minimum-response-time</code> • <code>one-minute-average-response-time</code> • <code>one-minute-maximum-response-time</code> • <code>one-minute-messages-sent</code> • <code>one-minute-messages-received</code> • <code>one-minute-messages-no-response</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/server/bindings/</code>	<p>Sensor for DHVPv4 server binding state statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name='routing-instance-name']/server/bindings/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>binding-state-v4server-binding</code> • <code>binding-state-v4server-init</code> • <code>binding-state-v4server-bound</code> • <code>binding-state-v4server-selecting</code> • <code>binding-state-v4server-requesting</code> • <code>binding-state-v4server-renew</code> • <code>binding-state-v4server-release</code> • <code>binding-state-server-restoring</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/ dhcp/v4/routing-instances/routing-instance/server/ statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for DHCPv4 telemetry for server statistics for a specific routing-instance.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics/</pre> <p>For example, the following resource path defines server statistics for the default:n000015k routing instance: /junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='n000015k']/server/statistics</p> <p>In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor /junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics/ the only value supported for <i>routing-instance-name</i> is default.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-v4server-total • dropped-v4server-bad-hware • dropped-v4server-bootp-pkt • dropped-v4server-bad-bootp-opcode • dropped-v4server-bad-options • dropped-v4server-bad-address • dropped-v4server-no-address • dropped-v4server-no-interface-cfg • dropped-v4server-no-local-address • dropped-v4server-short-pkt • dropped-v4server-no-bad-send • dropped-v4server-no-option60 • dropped-v4server-no-option82 • dropped-v4server-authentication • dropped-v4server-dynamic-profile • dropped-v4server-no-license • dropped-v4server-no-bad-dhcp-opcode • dropped-v4server-no-options • dropped-v4server-hop-limit • dropped-v4server-ttl-expired • dropped-v4server-bad_udp-checksum • dropped-v4server-inactive-vlan • dropped-v4server-era-start-ailed • dropped-v4server-client-lookup • dropped-v4server-lease-time-violation • offer-delayed • offer-delay-in-progress

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • offer-delay-total • msg-recv-v4server-boot-request • msg-recv-v4server-decline • msg-recv-v4server-discover • msg-recv-v4server-inform • msg-recv-v4server-release • msg-recv-v4server-request • msg-recv-v4server-renew • msg-recv-v4server-rebind • msg-recv-v4server-lease-query • msg-recv-v4server-bulklease-query • msg-sent-v4server-boot-reply • msg-sent-v4server-offer • msg-sent-v4server-boot-ack • msg-sent-v4server-nak • msg-sent-v4server-force-renew • msg-sent-v4server-unassigned • msg-sent-v4server-unknown • msg-sent-v4server-active • msg-sent-v4server-query-done
/junos/system/subscriber-management/client-protocols/ dhcp/v4/	<p>Sensor for DHCPv4 telemetry.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-total • dropped-bad-read • dropped-ip-header • dropped-short-packet • dropped-no-interface • dropped-no-routing-instance • dropped-no-memory • dropped-recovery-in-progress

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/ dhcp/v4/routing-instances/routing-instance/server/ statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for DHCPv4 server statistics</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance name:</p> <p>/junos/system/subscriber-management/protocols/dhcp/v4/routing-instances/routing-instance-name/server/statistics</p> <p>For example, the following resource path defines server statistics for the default: n000015k routing instance: /junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='n000015k']/server/statistics</p> <p>In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor /junos/system/subscriber-management/protocols/dhcp/v4/routing-instances/routing-instance-name/server/statistics/ the only value supported for <i>routing-instance-name</i> is default.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-v4server-total • dropped-v4server-bad-hware • dropped-v4server-bootp-pkt • dropped-v4server-bad-bootp-opcode • dropped-v4server-bad-options • dropped-v4server-bad-address • dropped-v4server-no-address • dropped-v4server-no-interface-cfg • dropped-v4server-no-local-address • dropped-v4server-short-pkt • dropped-v4server-no-bad-send • dropped-v4server-no-option60 • dropped-v4server-no-option82 • dropped-v4server-authentication • dropped-v4server-dynamic-profile • dropped-v4server-no-license • dropped-v4server-no-bad-dhcp-opcode • dropped-v4server-no-options • dropped-v4server-hop-limit • dropped-v4server-ttl-expired • dropped-v4server-bad_udp-checksum • dropped-v4server-inactive-vlan • dropped-v4server-era-start-ailed • dropped-v4server-client-lookup • dropped-v4server-lease-time-violation • offer-delayed • offer-delay-in-progress • offer-delay-total • msg-recv-v4server-boot-request

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • msg-recv-v4server-decline • msg-recv-v4server-discover • msg-recv-v4server-inform • msg-recv-v4server-release • msg-recv-v4server-request • msg-recv-v4server-renew • msg-recv-v4server-rebind • msg-recv-v4server-lease-query • msg-recv-v4server-bulklease-query • msg-sent-v4server-boot-reply • msg-sent-v4server-offer • msg-sent-v4server-boot-ack • msg-sent-v4server-nak • msg-sent-v4server-force-renew • msg-sent-v4server-unassigned • msg-sent-v4server-unknown • msg-sent-v4server-active • msg-sent-v4server-query-done

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/ dhcp/v4/routing-instances/routing-instance/relay/ statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for DHVPv4 relay binding state statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics/</pre> <p>For example, the following resource path defines relay statistics for the default:n000015k routing instance: <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='n000015k']/relay/statistics</code></p> <p>In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics/</code> the only value supported for the value <code>routing-instance-name</code> is default.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-v4relay-total • dropped-v4relay-bad-hardware • dropped-v4relay-bootp-packet • dropped-v4relay-bad-bootp-opcode • dropped-v4relay-bad-options • dropped-v4relay-bad-address • dropped-v4relay-no-address • dropped-v4relay-no-interface-cfg • dropped-v4relay-no-local-address • dropped-v4relay-short-packet • dropped-v4relay-bad-send • dropped-v4relay-option-60 • dropped-v4relay-relay-option • dropped-v4relay-option-82 • dropped-v4relay-authentication • dropped-v4relay-dynamic-profile • dropped-v4relay-dynamic-profile • dropped-v4relay-license • dropped-v4relay-bad-dhcp-opcode • dropped-v4relay-no-options • dropped-v4relay-hop-limit • dropped-v4relay-ttl-expired • dropped-v4relay-bad-udp-checksum • dropped-v4relay-inactive-vlan • dropped-v4relay-era-start-failed • dropped-v4relay-client-lookup • dropped-v4relay-proxy-no-server-addr • dropped-v4relay-lease-time-violation

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> dropped-v4relay-leasequery-repl-no-circuitid dropped-v4relay-leasequery-repl-with-error-code dropped-v4relay-leasequery-repl-with-query-term dropped-v4relay-older-leasequery-reply dropped-v4relay-abort-leasequery-reply-proc dropped-v4relay-during-leasequery-reply dropped-v4relay-relay-source-no-lpbk-interface v4relay-bootp-request-rcvd msg-recv-v4relay-decline msg-recv-v4relay-discover msg-recv-v4relay-inform msg-recv-v4relay-release msg-recv-v4relay-request msg-recv-v4relay-leaseactive msg-recv-v4relay-leaseunassigned msg-recv-v4relay-leaseunknown msg-recv-v4relay-leasequerydone v4relay-bootp-reply-rcvd msg-recv-v4relay-offer msg-recv-v4relay-ack msg-recv-v4relay-nak msg-recv-v4relay-forcerenew v4relay-bootp-reply-sent msg-sent-v4relay-offer msg-sent-v4relay-ack msg-sent-v4relay-nak msg-sent-v4relay-forcerenew msg-sent-v4relay-leasequery msg-sent-v4relay-bulkleasequery v4relay-bootp-request-sent msg-sent-v4relay-decline msg-sent-v4relay-discover msg-sent-v4relay-inform msg-sent-v4relay-release msg-sent-v4relay-request v4relay-bootp-forwarded-total v4relay-bootp-request-fwd v4relay-bootp-reply-fwd

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/relay/bindings/</code>	<p>Sensor for DHVPv6 relay binding state statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance name:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='routing-instance-name']/relay/bindings/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>binding-state-v6relay-binding</code> • <code>binding-state-v6relay-init</code> • <code>binding-state-v6relay-bound</code> • <code>binding-state-v6relay-selecting</code> • <code>binding-state-v6relay-requesting</code> • <code>binding-state-v6relay-renew</code> • <code>binding-state-v6relay-release</code> • <code>binding-state-relay-restoring</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
/junos/system/management/client-protocols/dhcp/v6/relay/servers/server/response-time	<p>Sensor for DHVPv6 server delay. The sensor periodically measures the minimum, average, and maximum delay or response time from the upstream DHCP server(s), as seen by the relay.</p> <p>DHCP relay does not track the state of the server. The no-response statistics are the difference between the messages sent and received during the measurement interval.</p> <p>The delay measurements are made over a 60-second measurement interval. Because the reporting interval can be as much as 59 seconds out of phase with the measurement interval, there is no design to align the response time values with the reporting interval.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='routing-instance-name']/relay/servers/server/response-time</pre> <p>The resource path can be refined to select a specific DHCP server by adding a server address filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/relay/servers/server[server-ip='server-ip']/response-time</pre> <p>The resource path can be refined to select a specific DHCP server in a specific routing instance by adding both a routing instance filter and a server filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='routing-instance-name']/relay/servers/server [server-ip='server-ip']/response-time</pre> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • one-minute-minimum-response-time • one-minute-average-response-time • one-minute-maximum-response-time • one-minute-messages-sent • one-minute-messages-received • one-minute-messages-no-response

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/server/bindings/</code>	<p>Sensor for DHVPv6 binding state statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <p><code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='routing-instance-name']/server/bindings/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>binding-state-v6server-binding</code> • <code>binding-state-v6server-init</code> • <code>binding-state-v6server-bound</code> • <code>binding-state-v6server-selecting</code> • <code>binding-state-v6server-requesting</code> • <code>binding-state-v6server-renew</code> • <code>binding-state-v6server-release</code> • <code>binding-state-server-restoring</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/ dhcp/v6/routing-instances/routing-instance/server/ statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for DHCPv6 server statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics/</pre> <p>For example, the following resource path defines server statistics for the default:n000015k routing instance: <code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='n000015k']/server/statistics</code></p> <p>In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics</code> the only value supported for <i>routing-instance-name</i> is default.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-v6server-total • dropped-v6server-no-routing-instance • dropped-v6server-bad-send • dropped-v6server-short-packet • dropped-v6server-bad-msgtype • dropped-v6server-bad-options • dropped-v6server-bad-srcaddress • dropped-v6server-relay-hop-count • dropped-v6server-bad-udp-checksum • dropped-v6server-no-client-id • dropped-v6server-strict-reconfigure • dropped-v6server-option-18 • dropped-v6server-authentication{ • dropped-v6server-dynamic-profile • dropped-v6server-license • dropped-v6server-inactive-vlan • dropped-v6server-era-start-failed • dropped-v6server-client-lookup • dropped-v6server-lease-time-violation • advertise-delayed • advertise-queued • advertise-total • msg-recv-v6server-dhcpv6-decline • msg-recv-v6server-dhcpv6-solicit • msg-recv-v6server-dhcpv6-information-request • msg-recv-v6server-dhcpv6-release • msg-recv-v6server-dhcpv6-request • msg-recv-v6server-dhcpv6-confirm

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none">• msg-recv-v6server-dhcpv6-renew• msg-recv-v6server-dhcpv6-rebind• msg-recv-v6server-dhcpv6-relay-forw• msg-recv-v6server-dhcpv6-leasequery• msg-sent-v6server-advertise• msg-sent-v6server-reply• msg-sent-v6server-logical_nak• msg-sent-v6server-reconfigure• msg-sent-v6server-relay-repl• msg-sent-v6server-leasequery-repl• msg-sent-v6server-leasequery-data• msg-sent-v6server-leasequery-done

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/ dhcp/v6/routing-instances/routing-instance/relay/ statistics/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor for DHVPv6 relay statistics.</p> <p>The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path:</p> <pre>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics/</pre> <p>For example, the following resource path defines relay statistics for the default:n000015k routing instance: <code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='n000015k']/relay/statistics</code></p> <p>In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics</code> the only value supported for <i>routing-instance-name</i> is default.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dropped-v6relay-total • dropped-v6relay-no-safd • dropped-v6relay-no-routing-instance • dropped-v6relay-bad-send • dropped-v6relay-short-packet • dropped-v6relay-bad-msgtype • dropped-v6relay-bad-options • dropped-v6relay-bad-srcaddress • dropped-v6relay-relay-hop-count • dropped-v6relay-bad-udp-checksum • dropped-v6relay-no-client-id • dropped-v6relay-strict-reconfigure • dropped-v6relay-relay-option • dropped-v6relay-option-18 • dropped-v6relay-option-37 • dropped-v6relay-authentication • dropped-v6relay-dynamic-profile • dropped-v6relay-license • dropped-v6relay-inactive-vlan • dropped-v6relay-era-start-failed • dropped-v6relay-client-lookup • dropped-v6relay-lease-time-violation • dropped-v6relay-leasequery-repl-no-client-data • dropped-v6relay-leasequery-repl-no-interfaceid • dropped-v6relay-leasequery-repl-with-client-link • dropped-v6relay-leasequery-repl-no-relay-data • dropped-v6relay-leasequery-repl-with-hop-cnt • dropped-v6relay-leasequery-repl-with-error-code

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • dropped-v6relay-leasequery-repl-with-query-term • dropped-v6relay-older-leasequery-reply • dropped-v6relay-abort-leasequery-reply-proc • dropped-v6relay-during-leasequery-reply • dropped-v6relay-relay-source-no-lpbk-interface • msg-recv-v6relay-decline • msg-recv-v6relay-solicit • msg-recv-v6relay-information-request • msg-recv-v6relay-release • msg-recv-v6relay-request • msg-recv-v6relay-confirm • msg-recv-v6relay-renew • msg-recv-v6relay-rebind • msg-recv-v6relay-relay-forw • msg-recv-v6relay-leasequery-repl • msg-recv-v6relay-leasequery-data • msg-recv-v6relay-leasequery-done • msg-recv-v6relay-advertise • msg-recv-v6relay-reply • msg-recv-v6relay-reconfigure • msg-recv-v6relay-relay-repl • msg-recv-v6relay-leasequery • msg-sent-v6relay-reply • msg-sent-v6relay-reconfigure • msg-sent-v6relay-relay-repl • msg-sent-v6relay-leasequery • msg-sent-v6relay-decline • msg-sent-v6relay-solicit • msg-sent-v6relay-information-request • msg-sent-v6relay-release • msg-sent-v6relay-request • msg-sent-v6relay-confirm • msg-sent-v6relay-renew • msg-sent-v6relay-rebind • msg-sent-v6relay-relay-forw • msg-sent-v6relay-leasequery-repl • msg-sent-v6relay-leasequery-data • msg-sent-v6relay-leasequery-done • v6relay-fwd-total • v6relay-fwd-request • v6relay-fwd-reply

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/client-protocols/l2tp/summary/</code>	<p>Sensor for L2TP telemetry information.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>l2tp-stats-total-tunnels</code> • <code>l2tp-stats-total-sessions</code> • <code>l2tp-stats-control-rx-packets</code> • <code>l2tp-stats-control-rx-bytes</code> • <code>l2tp-stats-control-tx-packets</code> • <code>l2tp-stats-control-tx-bytes</code>
<code>/junos/system/subscriber-management/client-protocols/ppp/statistics/</code>	<p>Sensors for PPP telemetry information.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>ppp-stats-total-subscriber-sessions</code> • <code>ppp-stats-sessions-disable-phase</code> • <code>ppp-stats-sessions-establish-phase</code> • <code>ppp-stats-sessions-network-phase</code> • <code>ppp-stats-sessions-authenticate-phase</code>
<code>/junos/system/subscriber-management/client-protocols/pppoe/statistics/</code>	<p>Sensors for PPPoE counts.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>padi-packets-sent</code> • <code>padi-packets-received</code> • <code>pado-packets-sent</code> • <code>pado-packets-received</code> • <code>padr-packets-sent</code> • <code>padr-packets-received</code> • <code>pads-packets-sent</code> • <code>pads-packets-received</code> • <code>service-error-sent</code> • <code>service-error-received</code> • <code>ac-error-sent</code> • <code>ac-error-received</code> • <code>generic-error-sent</code> • <code>generic-error-received</code> • <code>malformed-packets-received</code> • <code>unknown-packets-received</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/</code>	<p>Sensor for FPC resource statistics, including statistics for throttled sessions due to exceeding the line card load threshold (as measured by the routing engine to FPC round trip delay).</p> <p>The resource path can be refined to select a specific slot by adding a slot number filter to the resource path:</p> <p><code>/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc[slot='slot number']/statistics/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>heap-memory-used</code> • <code>client-session-denied-count</code> • <code>service-session-denied-count</code> • <code>rtt-throttled-sub-count-client</code> • <code>rtt-throttled-sub-count-service</code>
<code>/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/pfes/pfe</code>	<p>Sensor for FPC resource statistics at the Packet Forwarding Engine level. Periodically tracks line card statistics and Packet Forwarding Engine statistics.</p> <p>The resource path can be refined to select a specific Packet Forwarding Engine by adding a Packet forwarding Engine filter to the resource path:</p> <p><code>/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/pfes/pfe[pfe-no='pfe number']/</code></p> <p>The resource path can be refined to select a specific Packet Forwarding Engine by adding a slot number filter to the resource path:</p> <p><code>/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc[slot='slot number']/statistics/pfes/pfe[pfe-no='pfe number']/</code></p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>pfe-no</code> • <code>filter-memory-used</code> • <code>ifl-memory-used</code> • <code>expansion-memory-used</code> • <code>nh-memory</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/infra/network/dhcp/</code>	<p>Sensor for network stack DHCP. Periodically tracks packets processed by the BBE network stack to and from the DHCP application.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>rx-packet-cnt</code> • <code>era-drops</code> • <code>rx-no-connection</code> • <code>rx-malformed-cnt</code> • <code>rx-no-if-cnt</code> • <code>rx-ifl-invalid</code> • <code>rx-send-failed</code> • <code>tx-packet-cnt</code> • <code>packets-transmitted</code> • <code>tx-malformed-cnt</code> • <code>tx-null-pkt</code> • <code>tx-no-if-cnt</code> • <code>tx-no-iff-cnt</code> • <code>tx-no-rtt-cnt</code> • <code>tx-arp-failed</code> • <code>tx_arp_failed</code> • <code>tx-if-invalid</code> • <code>tx-send-failed</code> • <code>rx-while-not-connected</code>
<code>/junos/system/subscriber-management/infra/network/dvlan/</code>	<p>Sensor for network stack dynamic VLAN. Periodically maintains a count of the number of packets received that triggered dynamic VLAN interface creations.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>rx-packet-cnt</code>

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
/junos/system/subscriber-management/infra/network/io/	<p>Sensor for network stack IO. Periodically provides basic network stack input and output and tracks network stack packet statistics.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • l2-rx-packets-cnt • l2-rx-packets-failed • l2-rx-malformed-cnt • l2-rx-ifd-invalid • l2-rx-ifl-invalid • l2-rx-no-iff-cnt • l2-rx-if-create-failed • l2-bbe-io-rcv-l3-unknown-address-family • l2-rx-unsupported-inet-protocol • l2-rx-unsupported-inet6-protocol • l2-rx-unsupported-udp-protocol • l2-rx-unsupported-punt-af • l2-rx-v4-data-path-punt-pkt • l2-rx-v4-data-path-punt-pkt-drop • l2-rx-v6-data-path-punt-pkt • l2-rx-v6-data-path-punt-pkt-drop • l2-tx-packets-cnt • l2-tx-malformed-cnt • l2-tx-no-ifd-cnt • l2-tx-ifl-invalid • l2-bbe-io-send-tx-failed • l2-bbe-io-send-tx-failed-partial • l2-tx-v4-out-error-local-intf • l2-tx-v6-out-error-local-intf • l3-rx-packet-cnt • l3-rx-unsupported-protocol • l3-tx-packet-cnt • l3-tx-send-failed • l3-tx-v4-kernel-forward • l3-tx-v4-kernel-forward-drops • l3-tx-v6-kernel-forward • l3-tx-v6-kernel-forward-drops
/junos/system/subscriber-management/infra/network/dvlan/	<p>Sensor for network stack dynamic VLAN. Periodically maintains a count of the number of packets received that triggered dynamic VLAN interface creations.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • rx-packet-cnt

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
/junos/system/subscriber-management/infra/network/l2tp/	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor network stack L2TP. Periodically tracks L2TP packets processed by the BBE network stack to and from the L2TP application.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • rx-cnt • rx-pkt-cnt • ppp-rx-pkt-cnt • tx-pkt-cnt • ppp-rx-lcp-conf-req-count • ppp-rx-lcp-conf-ack-count • ppp-rx-lcp-conf-nack-count • ppp-rx-lcp-term-req-count • ppp-rx-lcp-term-ack-count • ppp-rx-lcp-echo-req-count • ppp-rx-lcp-echo-resp-count • ppp-rx-pap-req-count • ppp-rx-pap-ack-count • ppp-rx-pap-nack-count • ppp-rx-chap-challenge-count • ppp-rx-chap-resp-count • ppp-rx-chap-success-count • ppp-rx-chap-fail-count • ppp-rx-ipcp-conf-req-count • ppp-rx-ipcp-conf-ack-count • ppp-rx-ipcp-conf-nack-count • rx-malformed-cnt • ppp-rx-unknown-protocol • rx-msg-cnt • rx-msg-processd-cnt • rx-msg-err • rx-invalid-msg-cnt • tx-cnt • ppp-tx-lcp-conf-req-count • ppp-tx-lcp-conf-ack-count • ppp-tx-lcp-conf-nack-count • ppp-tx-lcp-echo-req-count • ppp-tx-lcp-echo-resp-count • ppp-tx-lcp-term-req-count • ppp-tx-lcp-term-ack-count • ppp-tx-pap-req-count • ppp-tx-pap-ack-count • ppp-tx-pap-nack-count • ppp-tx-chap-challenge-count

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none">• ppp-tx-chap-resp-count• ppp-tx-chap-success-count• ppp-tx-chap-fail-count• ppp-tx-ipcp-conf-req-count• ppp-tx-ipcp-conf-ack-count• ppp-tx-ipcp-conf-nack-count• ppp-tx-unknown-protocol• tx-pkt-send-failed• tx-pkt-err• tx-msg-cnt• tx-msg-err

Table 8: Broadband Edge gRPC Sensors *(continued)*

resource path	Description
<code>/junos/system/subscriber-management/infra/network/ ppp/</code>	

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<p>Sensor network stack PPP. Periodically tracks PPP packets processed by the BBE network stack to and from the PPP application.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> rx-network-pkt-cnt rx-plugin-pkt-cnt rx-lcp-conf-req-cnt rx-lcp-conf-ack-cnt rx-lcp-conf-nack-cnt rx-lcp-conf-rej-cnt rx-lcp-term-req-cnt rx-lcp-term-ack-cnt rx-lcp-code-rej-cnt rx-lcp-protocol-rej-cnt rx-lcp-echo-req-cnt rx-lcp-echo-reply-cnt rx-pap-req-cnt rx-pap-ack-cnt rx-pap-nack-cnt rx-chap-challenge-cnt rx-chap-resp-cnt rx-chap-success-cnt rx-chap-failure-cnt rx-ipcp-req-cnt rx-ipcp-ack-cnt rx-ipcp-nack-cnt rx-ipv6cp-req-cnt rx-ipv6cp-ack-cnt rx-ipv6cp-nack-cnt rx-malformed-cnt rx-no-if-cnt rx-unsupported tx-cnt tx-lcp-conf-req-cnt tx-lcp-conf-ack-cnt tx-lcp-conf-nack-cnt tx-lcp-echo-req-cnt tx-lcp-echo-reply-cnt tx-lcp-term-req-cnt tx-lcp-term-ack-cnt tx-pap-req-cnt tx-pap-ack-cnt tx-pap-nack-cnt

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
	<ul style="list-style-type: none"> • tx-chap-challenge-cnt • tx-chap-resp-cnt • tx-chap-success-cnt • tx-chap-failure-cnt • tx-ipcp-req-cnt • tx-ipcp-ack-cnt • tx-ipcp-nack-cnt • tx-ipv6cp-req-cnt • tx-ipv6cp-ack-cnt • tx-ipv6cp-nack-cnt • tx-unknown-pkt-cnt • tx-send-failed • tx-malformed-cnt
/junos/system/subscriber-management/infra/network/pppoe/	<p>Sensor for network stack PPPoE statistics. PPPoE packets processed by the BBE network stack to and from the PPPoE application are tracked.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • rx-cnt • rx-padi-cnt • rx-padr-cnt • rx-ppp-cnt • rx-malformed-cnt • rx-no-if-cnt • rx-unsupported • rx-padi-era-discards • tx-cnt • tx-send-failed
/junos/system/subscriber-management/infra/sdb/statistics/client-type/	<p>Sensor for session database resources session counts by client type.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • dhcp-client-count • vlan-client-count • ppp-client-count • pppoe-client-count • l2tp-client-count • static-client-count • vpls-pw-client-count • mlppp-client-count • essm-client-count • total-client-count

Table 8: Broadband Edge gRPC Sensors (*continued*)

resource path	Description
<code>/junos/system/subscriber-management/infra/sdb/statistics/state/</code>	<p>Sensor for session database resources tracking session counts by state.</p> <p>The following end paths are also supported for the resource path:</p> <ul style="list-style-type: none"> • <code>init-state-count</code> • <code>configured-state-count</code> • <code>active-state-count</code> • <code>terminating-state-count</code> • <code>terminated-state-count</code> • <code>total-state-count</code>

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, virtual MX Series (vMX) routers are also supported.
17.3R1	Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600 and EX9200 switches and the Routing PTX3000 routers are also supported.
17.3R1	Starting with Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensors are supported.
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics/</code> the only value supported is default .
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics/</code> the only value supported for <i>routing-instance-name</i> is default .
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics/</code> the only value supported for <i>routing-instance-name</i> is default .
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/server/statistics</code> the only value supported is default .
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor <code>/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='routing-instance-name']/relay/statistics</code> the only value supported is default .
17.2R1	Starting with Junos OS Release 17.2R1, QFX10000 switches, QFX5200 switches, and PTX1000 routers are supported.
16.1R3	Starting with Junos OS Release 16.1R3, the Junos Telemetry Interface supports gRPC remote procedure calls to subscribe to and receive telemetry data on MX Series routers and PTX3000 and PTX5000 routers.

Related Documentation

- [Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 58](#)

Understanding YANG on Devices Running Junos OS

YANG is a standards-based, extensible data modeling language that is used to model the configuration and operational state data, remote procedure calls (RPCs), and server event notifications of network devices. The NETMOD working group in the IETF originally designed YANG to model network management data and to provide a standard for the content layer of the Network Configuration Protocol (NETCONF) model. However, YANG is protocol independent, and YANG data models can be used independent of the transport or RPC protocol and can be converted into any encoding format supported by the network configuration protocol.

Juniper Networks provides YANG modules that define the Junos OS configuration hierarchy and operational commands and Junos OS YANG extensions. You can download the YANG modules from the Juniper Networks website, from the Juniper Networks GitHub repository for YANG, or you can generate the modules on the device running Junos OS.

YANG uses a C-like syntax, a hierarchical organization of data, and provides a set of built-in types as well as the capability to define derived types. YANG stresses readability, and it provides modularity and flexibility through the use of modules and submodules and reusable types and node groups.

A YANG module defines a single data model and determines the encoding for that data. A YANG module defines a data model through its data, and the hierarchical organization of and constraints on that data. A module can be a complete, standalone entity, or it can reference definitions in other modules and submodules as well as augment other data models with additional nodes.

A YANG module defines not only the syntax but also the semantics of the data. It explicitly defines relationships between and constraints on the data. This enables you to create syntactically correct configuration data that meets constraint requirements and enables you to validate the data against the model before uploading it and committing it on a device.

YANG uses modules to define configuration and state data, notifications, and RPCs for network operations in a manner similar to how the Structure of Management Information (SMI) uses MIBs to model data for SNMP operations. However, YANG has the benefit of being able to distinguish between operational and configuration data. YANG maintains compatibility with SNMP's SMI version 2 (SMIv2), and you can use libsmi to translate SMIv2 MIB modules into YANG modules and vice versa. Additionally, when you cannot use a YANG parser, you can translate YANG modules into YANG Independent Notation (YIN), which is an equivalent XML syntax that can be read by XML parsers and XSLT scripts.

You can use existing YANG-based tools or develop custom network management applications to utilize YANG modules for faster and more accurate network programmability. For example, a client application could leverage YANG modules to generate vendor-specific configuration data for different devices and validate that data before uploading it to the device. The application could also handle and troubleshoot unexpected RPC responses and errors.

For information about YANG, see [RFC 6020](#), *YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, and related RFCs.

- Related Documentation**
- [YANG Modules Overview](#)
 - [Using Juniper Networks YANG Modules](#)
 - [show system schema](#)

Configurable NETCONF Proxy for Junos Telemetry Interface

The Junos Telemetry Interface provides the capability to stream statistics for various defined attributes at a high scale. For a definition of these attributes, see “[Guidelines for gRPC Sensors \(Junos Telemetry Interface\)](#)” on page 70. Additionally, you can stream the result of NETCONF requests executed locally on the system at a lower scale by using configurable XML proxy. The NETCONF XML management protocol and Junos XML API fully document all options for every supported Junos OS operational request. This feature provides access to thousands of attributes available through NETCONF “get” remote procedure calls (RPCs). For more information about using these RPCs, see *Sending Requests to the NETCONF Server*.

Starting with Junos OS Release 17.3R1 on MX Series and PTX Series routers, you can add configurable, user-defined YANG files. YANG is a standards-based, extensible, hierarchical data modeling language that is used to model the configuration and state data used by NETCONF operations, RPCs, and server event notifications



BEST PRACTICE: Juniper Networks recommends that you not use YANG files that map to a extensive or verbose Junos OS operational commands, such as `show interfaces` or `show route`. The use of such a file could result in very slow or no streaming of telemetry data or very high CPU usage for various processes.

-
- [Creating a User-Defined YANG File on page 144](#)
 - [Example: Kernel Routing Table \(KRT\) Statistics on page 146](#)
 - [Installing a User-Defined YANG File on page 149](#)

Creating a User-Defined YANG File

To use the `xmlproxyd` process, also known as daemon, to translate telemetry data, create a `render.yang` file where the `dr:command-app` is set to `xmlproxyd`.



NOTE: The filename and module name included in the file must start with `xmlproxyd_`

- For the filename, add the extension `.yang`, for example, `xmlproxyd_ldpstats.yang`
- For the module name, use the filename without the extension `.yang`, for example, `xmlproxyd_ldpstats`

Custom YANG files for Junos OS conform to the YANG file syntax defined in RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*. Certain directives need to be present in the file that configure NETCONF proxy.

1. Provide a name for the module. It must start with `xmlproxy_`

For example, for a YANG File for Packet Forwarding Engine traffic statistics:

```
module xmlproxy_pfstatistics_oc {
...
```

2. Include the name of the process, also known as the daemon, that provides the operational state. For the Junos Telemetry Interface, you must always use `xmlproxyd`:

```
dr:command-app "xmlproxyd";
```

3. Specify the RPC for the NETCONF get request:

```
rpc juniper-netconf-get {
```

4. Specify the location of the output of the RPC:

```
dr:command-top-of-output "/junos";
```

5. Specify the command to execute the RPC:

```
dr:command-full-name "drend juniper-netconf-get";
```

6. Specify the CLI command to execute for retrieving data.

For example, to retrieve Packet Forwarding Engine traffic statistics:

```
dr:cli-command "show pfe statistics traffic";
```

7. Specify help for an RPC:

```
dr:command-help "default <get> rpc";
```

8. Specify the OpenConfig hierarchy and use the `dr:source` command to map to a container, a list, or a specific leaf.

For example, specify Packet Forwarding Engine statistics, including input packets and output packets:

```
output {
  container oc-pfe {
    dr:source "/pfe-statistics";
    list pfe-traffic {
      dr:source "pfe-traffic-statistics";
      leaf "pfe-ipackets" {
        type string;
        dr:source "pfe-input-packets"; // XML tag name
      }
      leaf "pfe-opackets" {
        type string;
        dr:source "pfe-output-packets"; // XML tag name
      }
    }
  }
}
```

9. The following example shows how to include these commands in a YANG file to enable the `xmlproxyd` process to retrieve the full operational state and render it in the OpenConfig format:

```
dr:command-app "xmlproxyd";
rpc juniper-netconf-get {
  dr:command-top-of-output "/junos";
  dr:command-full-name "drend juniper-netconf-get";
  dr:cli-command "show krt state";
  dr:command-help "default <get> rpc";
  output {
    container junos {
      container oc-pfe {
        dr:source "/pfe-statistics";
```

Example: Kernel Routing Table (KRT) Statistics

This example shows a YANG file created to stream KRT statistics.

```
/*
 * Example yang for generating OC equivalent of internal meta tree
 * save as "xmlproxyd_krtState.yang" on router.
 * cli : show krt state
 */

module xmlproxyd_krtState {
  yang-version 1;

  namespace "http://juniper.net/yang/software";

  prefix "krt";

  import drend {
    prefix dr;
  }
```



```
grouping krt-state-information-grouping {  
  
  list krt-queue-state {  
    key "operations-queued";  
    dr:source "krt-queue-state";  
    leaf operations-queued {  
      type uint32;  
      dr:source krtq-operations-queued;  
    }  
    leaf rt-table-adds {  
      dr:source krtq-rt-table-adds;  
      type uint32;  
    }  
    leaf interface-routes {  
      dr:source krtq-interface-routes;  
      type uint32;  
    }  
    leaf high-multicast-adds-changes {  
      dr:source krtq-high-multicast-adds-changes;  
      type uint32;  
    }  
    leaf top-indirect-adds-changes {  
      dr:source krtq-top-indirect-adds-changes;  
      type uint32;  
    }  
    leaf indirect-adds-changes {  
      dr:source krtq-indirect-adds-changes;  
      type uint32;  
    }  
    leaf indirect-deletes {  
      dr:source krtq-indirect-deletes;  
      type uint32;  
    }  
    leaf high-mps-adds {  
      dr:source krtq-high-mps-adds;  
      type uint32;  
    }  
    leaf high-mps-changes {  
      dr:source krtq-high-mps-changes;  
      type uint32;  
    }  
    leaf top-priority-adds {  
      dr:source krtq-top-priority-adds;  
      type uint32;  
    }  
    leaf top-priority-changes {  
      dr:source krtq-top-priority-changes;  
      type uint32;  
    }  
    leaf top-priority-deletes {  
      dr:source krtq-top-priority-deletes;  
      type uint32;  
    }  
    leaf high-priority-adds {  
      dr:source krtq-high-priority-adds;  
      type uint32;  
    }  
    leaf high-priority-changes {  
      dr:source krtq-high-priority-changes;  
      type uint32;  
    }  
  }  
}
```

```
leaf high-priority-deletes {
  dr:source krtq-high-priority-deletes;
  type uint32;
}
leaf normal-priority-indirects {
  dr:source krtq-normal-priority-indirects;
  type uint32;
}
leaf normal-priority-adds {
  dr:source krtq-normal-priority-adds;
  type uint32;
}
leaf normal-priority-changes {
  dr:source krtq-normal-priority-changes;
  type uint32;
}
leaf normal-priority-deletes {
  dr:source krtq-normal-priority-deletes;
  type uint32;
}
leaf least-priority-adds {
  dr:source krtq-least-priority-adds;
  type uint32;
}
leaf least-priority-changes {
  dr:source krtq-least-priority-changes;
  type uint32;
}
leaf least-priority-deletes {
  dr:source krtq-least-priority-deletes;
  type uint32;
}
leaf normal-priority-cnh-deletes {
  dr:source krtq-normal-priority-cnh-deletes;
  type uint32;
}
leaf normal-priority-gmp {
  dr:source krtq-normal-priority-gmp;
  type uint32;
}
leaf rt-table-deletes {
  dr:source krtq-rt-table-deletes;
  type uint32;
}
leaf operations-deferred {
  dr:source krtq-operations-deferred;
  type uint32;
}
leaf operations-canceled {
  dr:source krtq-operations-canceled;
  type uint32;
}
leaf async-count {
  dr:source krtq-async-count;
  type uint32;
}
leaf async-non-q-count {
  dr:source krtq-async-non-q-count;
  type uint32;
}
leaf time-until-next-run {
```

```

        dr:source krtq-time-until-next-run;
        type uint32;
    }
    leaf kernel-rt-learn {
        dr:source krtq-kernel-rt-learn;
        type uint32;
    }
}

dr:command-app "xmlproxyd";
rpc juniper-netconf-get {
    dr:command-top-of-output "/junos";
    dr:command-full-name "drend juniper-netconf-get";
    dr:cli-command "show krt state";
    dr:command-help "default <get> rpc";
    output {
        container junos {
            container krt-state-information {
                dr:source "/krt-state-information";
                uses krt-state-information-grouping;
            }
        }
    }
}
}
}

```

Installing a User-Defined YANG File

To add, validate, modify, or delete a user-defined YANG file for XML Proxy for the Junos Telemetry Interface, use the **request system yang** set of commands from the operational mode:

1. Specify the name of the YANG file and the file path to install. This command creates `.json` file in the `opt/lib/render` directory.

```
user@host> request system yang add package package-name proxy-xml module
file-path-name
```



NOTE: This command can be performed only on the current Routing Engine.

To add multiple YANG modules with the `request system yang add package package-name proxy-xml module` command, enclose the file-path-names in brackets: [*file-path-name 1 file-path-name 2*]

2. (Optional) Validate a YANG package you have added.

```
user@host> request system yang validate proxy-xml module file-path-name
```

3. (Optional) Update an existing YANG file that was previously added.

```
user@host> request system yang update package-name proxy-xml file-path-name
```

4. Delete an existing YANG file.

```
user@host> request system yang delete package-name
```

5. Verify that the YANG file has been installed by entering the **show system yang package** command.

```
user@host> show system yang package package-name
```

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1 on MX Series and PTX Series routers, you can add configurable, user-defined YANG files.

Related Documentation

- [Understanding YANG on Devices Running Junos OS on page 143](#)

request system yang add

Syntax `request system yang add package package-name <proxy-xml> module [modules]
 <action-script [scripts]>
 <translation-script [scripts]>
 <deviation-module [modules]>`

Release Information Command introduced in Junos OS Release 16.1 on MX Series and T Series routers. Command introduced in Junos OS Release 17.1 on EX Series and QFX Series switches and PTX Series routers. Command introduced in Junos OS Release 17.3 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances. **proxy-xml** option introduced in Junos OS Release 17.3 on MX Series and PTX Series routers.

Description Define a new YANG package with the modules, deviation modules, and scripts that are added to the device as part of the package, and merge the data models defined in the modules with the Junos OS schema. When you add a custom YANG data model to the device, you must also add at least one translation script or one action script, which provides the mapping between the new data model and Junos OS. To add multiple modules or scripts, include a space-delimited list of absolute or relative file paths enclosed in brackets.



NOTE: To install OpenConfig modules that are packaged as a compressed tar file, use the `request system software add` command. OpenConfig modules and scripts that are installed using the `request system software add` command are always associated with the package identifier `openconfig`.

When you create a new package, the device stores copies of the module and script files in a new location. The device also stores copies of the action script and translation script files under the `/var/db/scripts/action` and `/var/db/scripts/translation` directories, respectively. Junos OS validates the syntax of the modules and scripts, rebuilds its schema to include the new data models, and then validates the active configuration against this schema. Newly added RPCs and configuration hierarchies are immediately available for use.

Options **action-script [*scripts*]**—List of paths for one or more action scripts to add to the device as part of the package.

module [*modules*]—List of paths for one or more YANG modules to add to the device as part of the package. The device merges the data models defined in the modules with the Junos OS schema.

deviation-module [*modules*]—(Optional) List of paths for one or more modules that define deviation statements that should be applied to modules in the package.

package *package-name*—User-defined identifier that represents the collection of YANG modules and scripts.

proxy-xml module [*modules*]—List of paths for one or more new modules that provide user-defined OpenConfig mappings for the XML Proxy process to translate Junos Telemetry Interface statistics exported through gRPC into key-value pairs.

translation-script [*scripts*]—List of paths for one or more translation scripts to add to the device as part of the package.

Required Privilege Level

maintenance

Related Documentation

- *Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS*
- *Understanding the Management of Nonnative YANG Modules on Devices Running Junos OS*
- [Configurable NETCONF Proxy for Junos Telemetry Interface on page 144](#)
- [request system yang update on page 155](#)
- *show system yang package*

Sample Output

`request system yang add`

```
user@host> request system yang add package p1 module [yang/if.yang yang/if-aggregate.yang
yang/if-show.yang] deviation-module yang/deviation/if-devs.yang
translation-script translation/if.slax action-script action/if-show.py
```

```
YANG modules validation : START
YANG modules validation : SUCCESS
Scripts syntax validation : START
script check succeeds
Scripts syntax validation : SUCCESS
Scripts syntax validation : START
Scripts syntax validation : SUCCESS
TLV generation: START
TLV generation: SUCCESS
Building schema and reloading /config/juniper.conf.gz ...
Activating /config/juniper.conf.gz ...
mgd: commit complete
Restarting mgd ...
```

```
WARNING: cli has been replaced by an updated version:
CLI release 16.1R1 built by builder on 2016-03-30 13:46:11 UTC
Restart cli using the new version ? [yes,no] (yes) yes
```

```
Restarting cli ...
user@host>
```

request system yang delete

Syntax `request system yang delete package-name`

Release Information Command introduced in Junos OS Release 16.1 on MX Series and T Series routers. Command introduced in Junos OS Release 17.1 on EX Series and QFX Series switches and PTX Series routers. Command introduced in Junos OS Release 17.3 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.

Description Remove the given YANG package and all of its modules and scripts from the device, and remove the data models associated with that package from the Junos OS schema.



CAUTION: Before you delete a YANG package, ensure that the active configuration does not contain configuration data that has dependencies on the data models added by that package.



NOTE: You must use the `request system software delete` command to remove OpenConfig packages that were installed from a compressed tar file using the `request system software add` command.

When you delete a package, Junos OS rebuilds its schema to remove the data models associated with that package and then validates the active configuration against the newly updated schema. The device removes the copies of the module and script files that were generated when the package was created. The device also removes the copies of the package's action script and translation script files that are stored under the `/var/db/scripts/action` and `/var/db/scripts/translation` directories. If you downloaded the original module and script files to a different location, the original files remain unchanged.

Options `package-name`—Name of the YANG package to remove.

Required Privilege Level maintenance

Related Documentation

- *Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS*
- *Understanding the Management of Nonnative YANG Modules on Devices Running Junos OS*
- [request system yang add on page 151](#)
- `show system yang package`

Sample Output

request system yang delete

```
user@host> request system yang delete pl
Building schema and reloading /config/juniper.conf.gz ...
Activating /config/juniper.conf.gz ...
mgd: commit complete
Restarting mgd ...

WARNING: cli has been replaced by an updated version:
CLI release 16.1R1 built by builder on 2016-03-30 13:46:11 UTC

Restart cli using the new version ? [yes,no] (yes) yes

Restarting cli ...
```


request system yang update

Syntax	request system yang update <i>package-name</i> action-script [<i>scripts</i>] deviation-module [<i>modules</i>] module [<i>modules</i>] proxy-xml [<i>file-path-names</i>] translation-script [<i>scripts</i>]
Release Information	<p>Command introduced in Junos OS Release 16.1 on MX Series and T Series routers.</p> <p>Command introduced in Junos OS Release 17.1 on EX Series and QFX Series switches and PTX Series routers.</p> <p>Command introduced in Junos OS Release 17.3 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.</p> <p>proxy-xml option introduced in Junos OS Release 17.3 on MX Series and PTX Series routers.</p>
Description	<p>Update an existing YANG package to include new or modified YANG modules or scripts, and merge the updated data models in that package with the Junos OS schema.</p> <p>When you update a package, the device stores copies of the new and modified module and script files. Junos OS then rebuilds its schema to include the changes to the data models and validates the active configuration against this schema.</p>
Options	<p>package-name—Name of the YANG package to update.</p> <p>action-script [<i>scripts</i>]—List of paths for one or more action scripts to add to or update in the package.</p> <p>deviation-module [<i>modules</i>]—List of paths for one or more deviation modules to add to or update in the package.</p> <p>module [<i>modules</i>]—List of paths for one or more YANG modules to add to or update in the package.</p> <p>proxy-xml [<i>file-path-names</i>]—List of paths for one or more YANG modules to add to or update in the package that provide user-defined OpenConfig mappings for the XML Proxy process to translate Junos Telemetry Interface statistics exported through gRPC into key-value pairs.</p> <p>translation-script [<i>scripts</i>]—List of paths for one or more translation scripts to add to or update in the package.</p>
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> • <i>Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS</i> • Configurable NETCONF Proxy for Junos Telemetry Interface on page 144 • request system yang add on page 151 • <i>show system yang package</i>

Sample Output

request system yang update

```
user@host> request system yang update p1 module yang/if.yang

YANG modules validation : START
YANG modules validation : SUCCESS
TLV generation: START
TLV generation: SUCCESS
Building schema and reloading /config/juniper.conf.gz ...
Activating /config/juniper.conf.gz ...
mgd: commit complete
Restarting mgd ...

WARNING: cli has been replaced by an updated version:
CLI release 16.1R1 built by builder on 2016-03-30 13:46:11 UTC
Restart cli using the new version ? [yes,no] (yes) yes

Restarting cli ...
```

request system yang validate

Syntax	<code>request system yang validate action-script [<i>scripts</i>] module [<i>modules</i>] proxy-xml module [<i>modules</i>] translation-script [<i>scripts</i>]</code>
Release Information	<p>Command introduced in Junos OS Release 16.1 on MX Series and T Series routers.</p> <p>Command introduced in Junos OS Release 17.1 on EX Series and QFX Series switches and PTX Series routers.</p> <p>Command introduced in Junos OS Release 17.3 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.</p> <p>proxy-xml option introduced in Junos OS Release 17.3 on MX Series and PTX Series routers.</p>
Description	Validate the syntax of one or more YANG modules, translation scripts, or action scripts.
Options	<p>action-script <i>scripts</i>—List of paths for one or more action scripts to validate.</p> <p>module <i>modules</i>—List of paths for one or more YANG modules to validate.</p> <p>proxy-xml module <i>modules</i>—List of paths for one or more YANG modules to validate that provide user-defined OpenConfig mappings for the XML Proxy process to translate Junos Telemetry Interface statistics exported through gRPC into key-value pairs.</p> <p>translation-script <i>scripts</i>—List of paths for one or more translation scripts to validate.</p>
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> • <i>Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS</i> • <i>Understanding the Management of Nonnative YANG Modules on Devices Running Junos OS</i> • Configurable NETCONF Proxy for Junos Telemetry Interface on page 144

Sample Output

request system yang validate

```

user@host> request system yang validate module [yang/if.yang yang/if-aggregate.yang]
translation-script translation/if.slax
YANG modules validation : START
YANG modules validation : SUCCESS
Scripts syntax validation : START
script check succeeds
Scripts syntax validation : SUCCESS

```


PART 4

Best Practices

- [Best Practices for Implementing the Junos Telemetry Interface on page 161](#)

CHAPTER 7

Best Practices for Implementing the Junos Telemetry Interface

- [Guidelines for Specifying Data Reporting Intervals Junos Telemetry Interface on page 161](#)
- [Guidelines for Aggregating Junos Telemetry Interface Data on page 162](#)

Guidelines for Specifying Data Reporting Intervals Junos Telemetry Interface

The Junos Telemetry Interface enables you to provision sensors to collect and export data for various system resources without involving polling. A request to send data is sent once by a management station to stream periodic updates.

You can configure telemetry sensors to report data at a specified interval either through the command-line interface (CLI) or through the OpenConfig for Junos **telemetrySubscribe** remote procedure call (RPC). To configure using the CLI, include the **reporting-rate seconds** statement at the **[edit services analytics export-profile profile-name]** hierarchy level. For the **telemetrySubscribe** RPC, specify the sampling interval parameter, in milliseconds. In both cases, the interval specifies the amount of time between each subsequent export of data.

How to Determine the Reporting Interval for a System Resource

To determine the appropriate reporting interval for a specific system resource, follow these guidelines:

- Identify the required export interval for a given object, such as an interface.
- Identify the maximum number of objects reported by the sensor, such as the number of physical interfaces configured on a line card.
- Identify the minimum number of objects reported on each interval for a given sensor.
- Use the following formula to determine the best reporting interval:
 - $\text{Reporting interval} = \text{Required Export Interval Per Object} * \text{Minimum Number of objects reported on each Interval} / \text{Maximum Number of Objects}.$

Consider this example. There is a business requirement to report interface statistics every 30 seconds. At every interval, 10 interface records are reported, and the total number of interfaces is 96 for each line card. Using the reporting-interval formula, the reporting

interval should be 3.125 seconds. Currently, the reporting interval can be configured only as a multiple of 2, in seconds. Therefore, for this example, configure the reporting interval as 2 seconds in the CLI or 2000 milliseconds in the OpenConfig RPC.



TIP: The same metric might be reported more than once over a 30-second interval. For the purposes of effective visualization and data manipulation, it is quite common to aggregate data over fixed time spans.

Related Documentation • [Overview of the Junos Telemetry Interface on page 4](#)

Guidelines for Aggregating Junos Telemetry Interface Data

One important feature of the Junos Telemetry Interface is that data processing occurs at the collector that streams data, rather than the device. Data is not automatically aggregated, but it can be aggregated for analysis.

Data aggregation is useful in the following scenarios:

- Data for the same metric over fixed spans of time, such as, the average number physical interface ingress errors over a 30-second interval.
- Data from different sources (such as multiple line cards) for the same metric, such as label-switched path (LSP) statistics or filter counter statistics.
- Data from multiple sources, such as input and output statistics for aggregated Ethernet interfaces.

The follow sections describe how to perform data aggregation for various scenarios. The examples in these sections use the InfluxDB time-series database to accept queries on telemetry data. InfluxDB is an open source database written in Go specifically to handle time-series data.

Aggregating Data Over Fixed Time Spans

Aggregating data for the same metric over fixed spans of time is a common and useful way to detect trends. Metrics can include gauges, that is, single values, or cumulative counters. You might also want to aggregate data continuously.

Example: Aggregating Data for Gauge Metrics

In this example, data for

`JuniperNetworkSensors.jnpr_interface_ext.interface_stats.egress_queue_info.current_buffer_occupancy` from `port.proto` is written to the InfluxDB database with tags that identify the host name, an interface name and corresponding queue number and measurement called `current_buffer_occupancy`. See [Table 9 on page 163](#) for the specific values used in this example.

Table 9: Telemetry Data Values

Time Stamp (seconds)	Value	Tags
1458704133	1547	queue_number=0,interface_name='xe-1/0/0',host='sjc-a'
1458704143	3221	queue_number=0,interface_name='xe-1/0/0',host='sjc-a'
1458704155	4860	queue_number=0,interface_name='xe-1/0/0',host='sjc-a'
1458704166	6550	queue_number=0,interface_name='xe-1/0/0',host='sjc-a'

Each measurement data point has a timestamp and recorded value. In this example, the tag **queue_number** is the numerical identifier of the interface queue.

To aggregate this data over 30-second intervals, use the following influxDB query:

```
select mean(value) from current_buffer_occupancy
  where time >= $time_start and time <= $time_end and
         queue_number='0' and interface_name='xe-1/0/0' and host='sjc-a'
 group by time(30s)
```

For **\$time_start** and **\$time_end**, specify the actual range of time.

Example: Aggregating Data for Cumulative Statistics

Some Junos Telemetry Interface sensors report cumulative counter values, such as the number of ingress packets, defined as

JuniperNetworksSensors.jnpr_interface_ext.interface_stats.ingress_stats.packets.

It is common to derive traffic rates from packet or byte counters. Unlike with gauge metrics, the initial data point in the series for cumulative counters is used only to set the baseline.

Use the following guidelines to create a database query for cumulative statistics:

- Calculate the cumulative value for a specific time interval. You can calculate either an average among several data points recorded during the time interval, or you can interpolate a value. All data points should belong to the same series. If a counter reset has occurred between the two data points reported at different times, do not use both data points.
- Determine the appropriate value for the previous time interval. If a counter has been reset since the last update, declare that value as unavailable.
- If the previous interval is available, calculate the difference between the data points and the traffic rate.

These guidelines are summarized in the following influxDB query. This query assumes that data is stored in the measurement **ingress_packets**. The query uses the same tags as the gauge metric example as well as the tag for counter initialization time, **init_time**.

The query uses average values over a 30-second time interval. It calculates the rate for the metrics that have the same counter initialization.

```
select non_negative_derivative(mean(value)) from ingress_packets
  where time >= $time_start and time <= $time_end and
         interface_name='xe-1/0/0' and host='sjc-a'
  group by time(30s), init_time
```

Use the following query to calculate the number of packets received over an interval of time, without deriving the rate.

```
select difference(mean(value)) from ingress_packets
  where time >= $time_start and time <= $time_end and
         interface_name='xe-1/0/0' and host='sjc-a'
  group by time(30s), init_time
```

In some cases, more than one aggregated data point is returned by the query for a particular time interval. For example, four data points are available for a time interval. Two data points have `init_time t0`, and the other two have `init_time t1`. You can run a query that uses the last change timestamp tag, `last_change`, instead of `init_time`, to calculate the difference and to derive the rate between the two data points with the same last change timestamp.

```
select difference(mean(value)) from ingress_packets
  where time >= $time_start and time <= $time_end and
         interface_name='xe-1/0/0' and host='sjc-a'
  group by time(30s), last_change
```



TIP: These queries can all be run as continuous queries and can periodically populate new time-series measurements.

Aggregating Data From Multiple Sources

Certain metrics are reported from multiple line cards or packet forwarding engines. It is useful to aggregate data derived from different sources in the following scenarios:

- Packet and byte counts for label-switched paths (LSPs) are reported separately by each line card. However, a view of LSP paths for the entire device is required for path computation element controllers.
- For Juniper Networks devices that support virtual output queues, the tail drop or random early detection drop statistics for each queue are reported separately by each line card for every physical interface. It is useful to be able to aggregate the statistics for all the line cards for an interface.
- Filter counters for a firewall filter attached to a forwarding table or to an aggregated Ethernet interface are reported separately by each line card. It is useful to aggregate the statistics for all the line cards.

To aggregate data from multiple sources, perform the following:

1. Aggregate data for a specific period of time for each source, for example, each line card.
2. Aggregate the data you derive for each source in *step 1*.

For data stored in an InfluxDB database, you can complete *step 1* in the procedure by running a continuous query and populating a new measurement. We strongly recommend that you group the data points according to each source. For example, for LSP statistics, the **component_id** in the the gpb message identifies the line card sending the data. Group the data points based on each unique **component_id**.

Example: Aggregating Data from Multiple Sources

In this example, you run two queries to derive the LSP packet rate for data from all line cards.

First, you run the following continuous query on the measurement named **lsp_packet_count** for each **component_id** tag and the **counter_name** tag. Each unique **component_id** tag corresponds to a different line card. This query populates a new measurement, **lsp_packet_rate**.

```
select non_negative_derivative(mean(value)) as value from lsp_packet_count
into lsp_packet_rate
group by time(30s), component_id, counter_name, host
```



NOTE: The LSP statistics sensor does not report counter initialization time.

Use the new measurement derived from this continuous query—**lsp_packet_count**—to run the following query, which aggregates data from all line cards for packet rates for an LSP named **lsp-sjc-den-1**.

```
select sum(value) from lsp_packet_rate
where counter_name='lsp-sjc-den-1', host='sjc-a'
```



NOTE: Because this query does not group data according to the **component_id** tag, or line card, the LSP packet rates from all components, or line cards, are returned.

Aggregating Data for Multiple Metrics

It can be useful to aggregate metrics for multiple values. For example, for aggregated Ethernet interfaces, you would typically want to track packet and byte rates for each interface member as well as interface utilization for the aggregated link.

Example: Aggregating Multiple Metric Values

In this example, you run the following two queries:

- Continuous query to derive ingress packet counts for each member link in an aggregated Ethernet interface
- Query to aggregate packet count data for all the member links that belong to the same aggregated Ethernet interface

The following continuous query derives a measurement, **ingress_packets**, for each member link in an aggregated Ethernet interface. The **interface_name** tag identifies each member interface. You also use the **parent_ae_name** tag to identify membership in a specific aggregated Ethernet interface. Grouping each member link with the **parent_ae_name** tag ensures that data is collected only for current member links. For example, an interface might change its membership during the reporting interval. Grouping member interfaces with the specific aggregated Ethernet interface means that data for the member link will not be transferred to the new aggregated Ethernet interface of which it is now a member.

```
select difference(mean(value)) as value from ingress_packets
into ingress_packets_difference
group by time(30s), component_id, interface_name, host, parent_ae_name
```

The following query aggregates data for the ingress packets for the aggregated Ethernet interface, that is all member links.

```
select sum(value) from ingress_packets_difference
where parent_ae_name='ae0' and host='sjc-a'
```



NOTE: This query aggregates data for aggregated Ethernet interface ae0. The **parent_ae_name** tag does not verify the actual member links.

Related Documentation

- [Overview of the Junos Telemetry Interface on page 4](#)