Junos OS

Junos Telemetry Interface Feature Guide

Modified: 2019-06-25

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

Junos OS Junos Telemetry Interface Feature Guide

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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

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

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

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

GUI Conventions

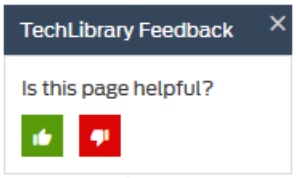
Table 2: Text and Syntax Conventions (continued)

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

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide 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 Juniper Care or Partner Support Services 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- 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 https://support.juniper.net/support/requesting-support/.

PART 1

Junos Telemetry Interface

CHAPTER 1

Understanding Junos Telemetry Interface

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, QFX10002, QFX10008, and QFX10016 switches, QFX5200 switches, and PTX1000 and PTX10008 routers are also supported. QFX5200 swtiches support only gRPC sensors.

Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600, EX4600-VC, 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, PTX10016 routers and virtual MX Series (vMX) routers are supported.

Starting with Junos OS Release 18.2R1, PTX10002 routers are also 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.

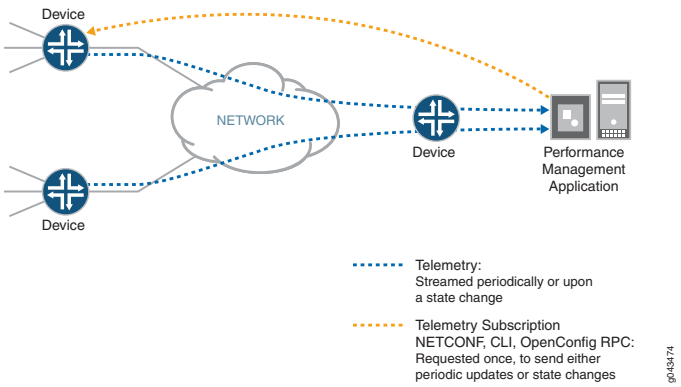
- An OpenConfig data model that generates data as gpb messages in a universal key/value format. OpenConfig for Junos OS, which you must download, supports the YANG data models. gRPC remote procedure calls (gRPC) are used to provision sensors and to subscribe to and receive telemetry data. gRPC is based 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.Starting in Junos OS Release 18.2R1, OpenConfig-based routing engine (RE) sensors can stream data as gpb structured messages over UDP.

Uses and Benefits

One 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.

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



Starting in Junos OS Release 18.1R1, a new sensor is available that allows syslog data to be streamed to network telemetry collector systems. Using the /junos/events/ sensor, and an export profile with a **reporting-rate** of 0, you can now stream event data along with statistical data to your telemetry-collection systems.

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
18.2R1	Starting with Junos OS Release 18.2R1, PTX10002 routers are also supported.
18.2R1	Starting in Junos OS Release 18.2R1, OpenConfig-based routing engine (RE) sensors can stream data as gpb structured messages over UDP.
18.1R1	Starting in Junos OS Release 18.1R1, a new sensor is available that allows syslog data to be streamed to network telemetry collector systems.
17.4R1	Starting with Junos OS Release 17.4R1, PTX10016 routers and virtual MX Series (vMX) routers are supported.
17.3R1	Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600, EX4600-VC, 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, QFX10002, QFX10008, and QFX10016 switches, QFX5200 switches, and PTX1000 and PTX10008 routers are also supported. QFX5200 swtiches 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 8
- Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32

CHAPTER 2

Native Sensors for Junos Telemetry Interface

Understanding the Junos Telemetry Interface Export Format of Collected Data

The Junos Telemetry Interface supports two ways of exporting data in the 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 senors 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 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, Junos Telemetry Interface is supported on the Routing Control and Board (RCB) on PTX3000 routers, QFX5110 switches, and EX4600 and EX9200 switches.

Starting with Junos OS Release 17.4R1, MX2008 routers are supported.

- Understanding the Sensor Data Encapsulation Format on page 9

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 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 12 For a 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: https://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)**.

Protocol buffers message Definition

Following is the message definition for **TelemetryStream** in the 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 increasesing 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 11 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 the interval over which the rate is collected.
Average	The average of several samples of a base metric. For example, an *average queue depth* 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 *peak queue depth* 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
17.4R1	Starting with Junos OS Release 17.4R1, MX2008 routers are supported.
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, 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 20

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.

Starting with Junos OS Release 19.1R1, MX Series routers operating with MS-MIC and MS-MPC, QFX10002 switches, and PTX10002 routers are also supported.

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 13
- Configuring a Streaming Server Profile on page 16
- Configuring a Sensor Profile on page 17
- Verifying Junos Telemetry Interface Sensor Configuration on page 19

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**:

   ```
   [edt 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                    : 20
```

```
    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
19.1R1	Starting with Junos OS Release 19.1R1, MX Series routers operating with MS-MIC and MS-MPC, QFX10002 switches, and PTX10002 routers are also supported.
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.

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: https://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:
"S1:/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 12

CHAPTER 3

OpenConfig and gRPC for Junos Telemetry Interface

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.

Starting with Junos OS Release 17.4R1, MX2008 routers are supported.

Starting with Junos OS Release 18.3R1, ON_CHANGE streaming of LLDP telemetry sensor information is supported through gRPC for MX Series and PTX Series routers.

Starting with Junos OS Release 18.3R1, QFX5120-AY and EX4650 switches are also supported.

Starting with Junos OS Release 18.4R1, EX4600 switches are also supported.

Starting with Junos OS Release 18.4R1, MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers are also supported.

Starting with Junos OS Release 19.1R1, MX Series routers operating with MS-MIC and MS-MPC, QFX10002 switches, and PTX10002 routers are also supported.

Starting in Junos OS Evolved Release 19.1R1, OpenConfig (OC) and Junos Telemetry Interface (JTI) are supported. Both gRPC APIs and the customer-facing CLI remain the same as for the Junos OS. As was standard for Junos OS, Network Agent (NA) and OC packages are part of the Junos OS Evolved image.

Starting with Junos OS Evolved 19.1R1, Packet Forwarding Engine sensors on PTX10003 routers are also supported.

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 45.

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: https://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.

NOTE: Starting in Junos OS Release 18.2R1, when an external streaming server, or collector, provisions sensors to export data through gRPC on devices running Junos OS, the sensor configuration is committed to the **junos-analytics** instance of the ephemeral configuration database, and the configuration can be viewed by using the **show ephemeral-configuration instance junos-analytics** operational command. In earlier releases, the sensor configuration is committed to the default instance of the ephemeral configuration database.

See Table 4 on page 34 for a list and descriptions of the RPCs implemented to the support the Junos Telemetry Interface.

Table 4: 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 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.

NOTE: Starting in Junos OS Release 18.2R1, data streamed through gRPC can be formatted as protobuf in addition to key/value pairs for OpenConfig-based routing engine (RE) sensors. These sensors are in addition to the packet forwarding engine (PFE) sensors.

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.

Exporting Packet Forwarding Engine Traffic Sensor Data

Starting with Junos OS Release 17.4R1, you can export Packet Forwarding Engine traffic statistics through the Junos Telemetry Interface for MX Series and PTX Series routers. Both UDP and gRPC are supported.

This sensor tracks reporting of Packet Forwarding Engine statistics counters and provides visibility into Packet Forwarding Engine error and drop statistics. The resource name for the sensor is **/junos/system/linecard/packet/usage/**. The OpenConfig paths report data specific to CPU, NPU and center chip (CC). The following paths are supported:

- **/components/component[name='FPC*id*:NPU*id*']/properties/property[name='*counter*']/state/value**, where FPC refers to the Flexible PIC Concentrator and NPU refers to the network processing unit (packet forwarding engine). A sample resource path is **/components/component[name='FPC0:NPU3']/properties/property[name='ts-output-pps']/state/value** where **hwds-data-error** is the counter for **Hardware Discards: Data Error**.

- **/components/component[name='FPC*id*:CC*id*']/properties/property[name='*counter*']/state/value**, where FPC refers to the Flexible PIC Concentrator and CC refers to the center chip. A sample resource path is **/components/component[name='FPC0:CC1']/properties/property[name='lpbk-packets"]/state/value** where **lpbk-packets** is the count of **Forward packets** specific to FPC0, center chip 1.
- **/components/component[name='FPC*id*']/properties/property[name='*counter*']/state/value**, where FPC refers to the Flexible PIC Concentrator. A sample resource path is **/components/component[name='FPC0']/properties/property[name='lts-input-packets"]/state/value** where **lts-input-packets** is the CPU counter **Local packets input**.

To provision the sensor to export data through gRPC, use the **telemetrySubcribe** RPC to specify telemetry parameters. For streaming through UDP, all parameters are configured at the **[edit services analytics]** hierarchy level.

The following is a map of counters to output fields in the **show pfe statistics traffic** command or **show pfe statistics traffic detail** command (supported only on MX Series routers).

```
CPU stats: (FPCX:CPUY)
Packet Forwarding Engine local traffic statistics:
    Local packets input                :                    2
    Local packets output               :                    1
    Software input control plane drops  :                   0
    Software input high drops          :                    0
    Software input medium drops        :                    0
    Software input low drops           :                    0
    Software output drops              :                    0
    Hardware input drops               :                    0


Counter
lts-input-packets             Local packets input
lts-output-packets            Local packets output
lts-sw-input-control-drops    Software input control plane drops
lts-sw-input-high-drops       Software input high drops
lts-sw-input-medium-drops     Software input medium drops
lts-sw-input-low-drops        Software input low drops
lts-sw-output-low-drops       Software output drops


NPU stats: (FPCX:CCY)
    Input  packets:                 1169                    0 pps
    Output packets:                    0                    0 pps
    Fabric Input  :            277235149               16078 pps
    Fabric Output :            277235149               16079 pps


Counter
ts-input-packets               Input  packets
ts-input-packets-pps           Input  packets in pps
ts-output-packets              Output packets
ts-output-packets-pps          Output packets in pps
ts-fabric-input-packets        Fabric Input
ts-fabric-input-packets-pps    Fabric Input in pps
ts-fabric-output-packets       Fabric Output
```

```
ts-fabric-output-packets-pps   Fabric Output in pps

Packet Forwarding Engine loopback statistics:
    Forward packets :                   0                   0 pps
    Forward bytes   :                   0                   0 bps
    Drop packets    :                   0                   0 pps
    Drop bytes      :                   0                   0 bps


Counter
lpbk-packets                  Forward packets
lpbk-packets-pps              Forward packets pps
lpbk-packets-byte             Forward bytes
lpbk-packets-bps              Forward bytes   bps

lpbk-drop-packets             Drop packets
lpbk-drop-packets             Drop packets pps
lpbk-drop-packets             Drop bytes
lpbk-drop-packets             Drop bytes bps



Lu chips stats: FPCx:NPUY
Counter
lts-hw-input-drops
hwds-normal           Hardware discards normal discard
hwds-fabric           Hardware discards fabric drops
hwds-info-cell        Hardware discards info cell drops
hwds-timeout          Hardware discards timeour
hwds-truncated-key    Hardware discards truncated key
hwds-bits-to-test     Hardware discards bits to test
hwds-stack-underflow  Hardware discards stack underflow
hwds-stack-overflow   Hardware discards stack overflow
hwds-data-error       Hardware discards data error
hwds-extended         Hardware discards extended discard
hwds-invalid-iif      Hardware discards invalid interface
hwds-input-checksum   Hardware discards input checksum
hwds-output-mtu
hwds-inet-bad-route
hwds-inet6-bad-route
hwds-filter-discard
hwds-dlu-not-routable
```

Enabling “ON CHANGE” Sensor Support Through Network Management Interface (gNMI)

Periodical streaming of OpenConfig operational states and counters has been supported since Junos OS Release 16.1, exporting telemetry data from Juniper equipment to an external collector. While useful in collecting all the needed information and creating a baseline “snapshot,” periodical streaming is less useful for time-critical missions. In such instances, you can configure ON_CHANGE streaming for an external collector to receive information only when operational states experience a change in state.

To support ON_CHANGE streaming, a new specification called gRPC Network Management Interface (gNMI) is implemented for the modification and retrieval of configurations from a network element. Additionally, the gNMI specification can be used to generate and control telemetry streams from a network element to a data collection system. Using the new gNMI specification, one gRPC service definition can provide a

single implementation on a network element for both configuration and telemetry as well as a single NMS element to interact with a device by means of telemetry and configuration RPCs.

The Junos file package (junos-telemetry-interface) includes the gnmi.proto file and GnmiJuniperTelemetryHeader.proto Juniper extension for gNMI support.

Information about the RPCs supporting this feature can be found in the gNMI Proto file version 0.4.0 (the supported version) and the specification released

- https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md
- https://github.com/openconfig/gnmi/blob/master/proto/gnmi/gnmi.proto

The telemetry RPC **subscribe** under gNMI service supports ON_CHANGE streaming. RPC **subscribe** allows a client to request the target to send it values of particular paths within the data tree. Values may be streamed (STREAM), sent one-off on a long-lived channel (POLL), or sent one-off as a retrieval (ONCE).

If a subscription is made for a top level container with a sample frequency of 0, leaves with ON_CHANGE support are streamed based on events. Other leaves will not be streamed.

NOTE: In order to permit a device to decide which nodes will be streamed as ON_CHANGE and which will SAMPLE, the collector must subscribe for TARGET_DEFINED with sample_interval.

Enabling Client Streaming and Bidirectional Streaming of Telemetry Sensor Information

Starting with Junos OS Release 18.1R1, OpenConfig support through Remote Procedure Calls (gRPC) and JTI is extended to support client streaming and bidirectional streaming of telemetry sensor information on MX Series and PTX Series routers.

APIs are implemented in Junos based on Protobuf specifications for OpenConfig. These APIs perform configuration, operational state retrieval, and telemetry on Junos routers using gRPC as the transport mechanism.

With client streaming, the client sends a stream of requests to the server instead of a single request. The server typically sends back a single response containing status details and optional trailing metadata. With bidirectional streaming, both client and server send a stream of requests and responses. The client starts the operation by invoking the RPC and the server receives the client metadata, method name, and deadline. The server can choose to send back its initial metadata or wait for the client to start sending requests. The client and server can read and write in any order. The streams operate completely independently.

Junos devices can be managed through API (RPC) prototypes:

- rpc Capabilities (CapabilityRequest)

  Returns (CapabilityResponse). Allows the client to retrieve the set of capabilities that is supported by the target.

- rpc Get (GetRequest)

  Returns (GetResponse). Retrieves a snapshot of data from the target.

- rpc Set (SetRequest)

  Returns (SetResponse). Allows the client to modify the state of data on the target.

- rpc Subscribe (stream SubscribeRequest)

  Returns (stream SubscribeResponse). Allows a client to request the target to send it values for particular paths within the data tree. These values may be streamed (STREAM) or sent one-off on a long-lived channel (POLL), or sent as a one-off retrieval (ONCE). If a subscription is made for a top-level container with a sample frequency of 0, leaves with ON_CHANGE support are streamed based on events. Other leaves will not be streamed.

Juniper Extension Toolkit (JET) support provides insight to users regarding the status of clients connected to JSD. JET support for gRPC includes expanding the maximum number of clients that can connect to JSD from 8 to 30 (the default remains 5). To specify the maximum number of connections, include the **max-connections** statement at the [**edit system services extension-service request-response grpc**] hierarchy level.

To provide information regarding the status of clients connected to JSD, issue the enhanced **show extension-service client information** command and include the **clients** or **servers** options. The **clients** option displays request-response client information. The **servers** option displays request-response server information.

Enabling Streaming of Telemetry Sensor Information for SR-TE policies (BGP or Static)

Starting with Junos OS Release 18.3R1, OpenConfig support through gRPC and JTI provides continuous statistics streaming via the same sensor irrespective of the route that is active (BGP or static) for a given Segment Routing Traffic Engineering (SR-TE) policy.

This feature provides support for BGP [DRAFT-SRTE] and statically configured SR-TE policies at ingress routers.

To provision the sensor to export data through gRPC streaming, use the telemetrySubscribe RPC to specify telemetry parameters. Include the resource path **/mpls/signaling-protocols/segment-routing/** to export these statistics.

In addition to configuring the sensor, you must enable statistics collection through the Junos OS. To do this, include the **statistics** configuration statement at the [**edit protocols source-packet-routing telemetry**] hierarchy level. Optionally, you can limit statistics by including the **no-transit** or **no-ingress** parameter.

See "Configure a Telemetry Sensor in Junos" on page 177 for instructions on configuring a sensor.

See "Guidelines for gRPC Sensors (Junos Telemetry Interface)" on page 51 for further information about resource paths.

Support for LSP Statistics

You can provision the LSP statistics sensor **/junos/services/label-switched-path/usage/** to monitor per-MPLS LSP statistics. Telemetry data is streamed from Junos devices and exported through JTI to external collectors at configurable intervals through gRPC without involving polling.

Initial support of this feature in Junos OS Release 15.1F6 supported ingress LSPs only when a subscription was made to **/junos/services/label-switched-path/usage/**. With bypass support added to this feature in Junos OS Release 17.4R1, this subscription now streams both ingress LSP and bypass LSP statistics to a collector.

Statistics that are streamed are similar to the output displayed by the operational mode commands **show mpls lsp bypass statistics** and **show mpls lsp ingress statistics**.

For bypass LSPs, the following are exported:

- 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.

When the bypass LSP is active, traffic is exported both on the bypass LSP and the ingress (protected) LSP.

To provision a sensor to export data through gRPC, use the telemetrySubscribe RPC to specify telemetry parameters. Streaming telemetry data through gRPC also requires the OpenConfig for Junos OS module. Both OpenConfig and Network Agent packages are bundled into the Junos OS image by default.

See "Configuring a Junos Telemetry Interface Sensor (CLI Procedure)" on page 12 for information about configuring a UDP (native) sensor.

See Table 5 on page 40 for the level of LSP sensor support by platform.

Table 5: LSP Support by Platform

Platform	Ingress LSP, UDP Feature Introduced
MX80/MX104	Junos OS Release 15.1F6 Junos OS Release 16.1R3 Junos OS Release 17.2R1
MX Series with MPC	Junos OS Release 15.1F6

Table 5: LSP Support by Platform (continued)

Platform	Ingress LSP, UDP Feature Introduced
PTX5000 with FPC3	
PTX3000 with FPC3	Junos OS Release 15.1F6 Junos OS Release 16.1R3 Junos OS Release 17.2R1
PTX Series with FPC1/2	Junos OS Release 15.1F6 Junos OS Release 16.1R3 Junos OS Release 17.2R1
PTX1000	Junos OS Release 16.1R3
PTX10000	Junos OS Release 17.3R1
PTX10001-20C	
PTX10002	Junos OS Release 19.1R1
VMX	Junos OS Release 17.3R1
MX150	Junos OS Release 17.4R1
EX4600	Junos OS Release 18.4R1
EX4650	Junos OS Release 18.3R1
EX9200	Junos OS Release 17.3R1
QFX10000	
QFX5200	Junos OS Release 17.2R1
QFX10002	Junos OS Release 19.1R1
QFX5100	Junos OS Release 18.2R1
QFX5110	Junos OS Release 18.2R1

Table 5: LSP Support by Platform (continued)

Platform	Ingress LSP, UDP Feature Introduced
QFX5120-48Y	Junos OS Release 18.3R1
QFX5200	Junos OS Release 18.2R1

Release History Table

Release	Description
19.1R1 EVO	Starting in Junos OS Evolved Release 19.1R1, OpenConfig (OC) and Junos Telemetry Interface (JTI) are supported. Both gRPC APIs and the customer-facing CLI remain the same as for the Junos OS. As was standard for Junos OS, Network Agent (NA) and OC packages are part of the Junos OS Evolved image.
19.1R1	Starting with Junos OS Release 19.1R1, MX Series routers operating with MS-MIC and MS-MPC, QFX10002 switches, and PTX10002 routers are also supported.
19.1R1	Starting with Junos OS Evolved 19.1R1, Packet Forwarding Engine sensors on PTX10003 routers are also supported.
18.4R1	Starting with Junos OS Release 18.4R1, MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers are also supported.
18.3R1	Starting with Junos OS Release 18.3R1, ON_CHANGE streaming of LLDP telemetry sensor information is supported through gRPC for MX Series and PTX Series routers.
18.3R1	Starting with Junos OS Release 18.3R1, QFX5120-AY and EX4650 switches are also supported.
18.3R1	Starting with Junos OS Release 18.4R1, EX4600 switches are also supported.
18.2R1	Starting in Junos OS Release 18.2R1, when an external streaming server, or collector, provisions sensors to export data through gRPC on devices running Junos OS, the sensor configuration is committed to the **junos-analytics** instance of the ephemeral configuration database, and the configuration can be viewed by using the **show ephemeral-configuration instance junos-analytics** operational command.
18.1R1	Starting with Junos OS Release 18.1R1, OpenConfig support through Remote Procedure Calls (gRPC) and JTI is extended to support client streaming and bidirectional streaming of telemetry sensor information on MX Series and PTX Series routers.
18.1R1	Starting with Junos OS Release 18.3R1, OpenConfig support through gRPC and JTI provides continuous statistics streaming via the same sensor irrespective of the route that is active (BGP or static) for a given Segment Routing Traffic Engineering (SR-TE) policy.
17.4R1	Starting with Junos OS Release 17.4R1, MX2008 routers are supported.
17.4R1	Starting with Junos OS Release 17.4R1, you can export Packet Forwarding Engine traffic statistics through the Junos Telemetry Interface for MX Series and PTX Series routers. Both UDP and gRPC are supported.
17.4R1	With bypass support added to this feature in Junos OS Release 17.4R1, this subscription now streams both ingress LSP and bypass LSP statistics to a collector.
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.
15.1F6	Initial support of this feature in Junos OS Release 15.1F6 supported ingress LSPs only when a subscription was made to **/junos/services/label-switched-path/usage/**.

Related Documentation

- Installing the Network Agent Package (Junos Telemetry Interface) on page 45
- *Release Information for Junos OS with Upgraded FreeBSD*
- Guidelines for gRPC Sensors (Junos Telemetry Interface) on page 51
- *statistics*
- *telemetry*

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.

Starting in Junos OS Release 18.3R1, the Junos OS image includes the Network Agent. You do not need to install Network Agent separately. This is true for Junos OS with upgraded FreeBSD and legacy Junos OS.

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. The alue is **32** for 32-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.**

An example of a valid Network Agent package name is:

- **network-agent-x86-32-16.1R4.12-C1.1.tgz**

Use the 32-bit Network Agent package for both 32-bit and 64-bit versions of Junos OS or Junos OS Evolved.

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: https://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: https://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-agent** 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-agent-x86-32-16.1R3.16-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 48.

Release History Table

Release	Description
18.3R1	Starting in Junos OS Release 18.3R1, the Junos OS image includes the Network Agent.
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 32

gRPC Services for Junos Telemetry Interface

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 45.

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
   ```

   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 or you can rely on the default IP address (::) to enable Junos to "listen" for all IPv4 and IPv6 addresses on incoming connections. If you would rather specify an IP address, follow stp b. below.

   a. Specify a local certificate-name. The certificate can be any user-defined value from the certificate configuration (not shown here). The certificate name should used in thisexample is **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 32767 for accepting incoming connections through gRPC.

NOTE: Port 32767 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 32767
```

See Also

- Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32
- *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 48.

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 extention-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-verfiy.**

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 extention-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.

See Also

- ssl on page 275

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 JunosOS Release 17.2R1, QFX10002, QFX10008, and QFX10016 switches, QFX5200 switches, and PTX1000 and PTX10008 routers are also supported.

Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600, EX4600-VC, 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 18.2R1, PTX10002 routers are also supported.

Starting with Junos OS Release 17.4R1, PTX10016 routers and virtual MX Series (vMX) routers are also supported.

Starting with Junos OS Release 18.1R1, QFX5210-64C switches and QFX5100 switches are also supported.

Starting with Junos OS Release 18.1R1, ON_CHANGE streaming of ARP, ND, and IP sensor information associated with interfaces is supported through gRPC for MX Series routers and PTX Series routers.

Starting with Junos OS Release 18.3R1, ON_CHANGE streaming of LLDP telemetry sensor information is supported through gRPC for MX Series and PTX Series routers.

Starting with Junos OS Release 18.3R1, QFX5120-AY and EX4650 switches are also supported.

Starting with Junos OS Release 18.4R1, EX4600 switches are also supported.

Starting with Junos OS Release 18.4R1, MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers are also supported.

Starting in Junos OS Evolved Release 19.1R1, OpenConfig (OC) and Junos Telemetry Interface (JTI) are supported. Both gRPC APIs and the customer-facing CLI remain the same as for the Junos OS. As was standard for Junos OS, Network Agent (NA) and OC packages are part of the Junos OS Evolved image.

Starting with Junos OS Evolved 19.1R1, Packet Forwarding Engine sensors on PTX10003 routers are also supported.

See Table 6 on page 53 for information about which sensors are supported with gRPC and on which platforms.

See Table 7 on page 130 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 6 on page 53 for a description of supported gRPC sensors and Table 7 on page 130 for a description of supported broadband edge (BBE) gRPC sensors, including the subscription path you use to provision the sensors.

Table 6: gRPC Sensors

resource path	Description
/components/component/	Sensor for chassis components. ON_CHANGE notification is triggered if a component (FPC) is inserted or removed or if a component's power is on or off (FPC is online or offline). Instant reporting of such events is handled with this sensor. ON_CHANGE streaming is supported on MX960, MX2010, MX2020, PTX-5000, PTX1000, and PTX10000 routers starting with Junos OS Release 18.4R1. You can also add the following endpoints to the path to stream specific statistics: • name • state/id • state/description • state/serial-no • state/part-no • state/type-Identifies the ON_CHANGE event type. Event types are: **FRU_ADD**, **FRU_REMOVE**, **FRU_POWERON**, and **FRU_POWEROFF**. For more information about ON_CHANGE streaming, see "Understanding OpenConfig and gRPC on Junos Telemetry Interface" on page 32.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component/subcomponents/subcomponent[name='FPC*id*:NPU*id*']/properties/property/[name=' *counter*']/state/value	Sensor for packet forwarding engine statistics. The subcomponent name ***npu-id*** refers to the number of the packet forwarding engine. This sensor provides visibility into packet forwarding engine errors and drops. Supported on MX Series routers and PTX Series routers starting with Junos OS Release 17.4R1.MX960|MX2010|MX2020|PTX-5000|PTX1000|PTX10000 The value for ***counter*** is one of the following; • lts-hw-input-drops • hwds-normal • hwds-fabric • hwds-info-cell • hwds-timeout • hwds-truncated-key • hwds-bits-to-test • hwds-stack-underflow • hwds-stack-overflow • hwds-inet6-bad-route • hwds-inet-bad-route • hwds-filter-discard • hwds-dlu-not-routable • hwds-data-error • hwds-extended • hwds-invalid-iif • hwds-input-checksum • hwds-output-mtu • lts-input-packets • lts-output-packets • lts-sw-input-control-drops • lts-sw-input-high-drops • lts-sw-input-medium-drops • lts-sw-input-low-drops • lts-sw-output-low-drops

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component/subcomponents/subcomponent[name='FPC*ID*:CC*id*']/properties/property/[name=' *counter*']/state/value	Sensor for packet forwarding engine statistics. The subcomponent name ***cc-id*** refers to the center chip. This sensor provides visibility into packet forwarding engine errors and drops. Supported on MX Series routers and PTX Series routers starting with Junos OS Release 17.4R1. The value for ***counter*** is one of the following; • **ts-fabric-input-pps** • **ts-fabric-output-pps** • **ts-fabric-input-packets** • **ts-fabric-output-packets** • **lpbk-packets** • **lpbk-pps** • **lpbk-bytes** • **lpbk-pps** • **lpbk-drop-packets** • **lpbk-drop-pps** • **lpbk-drop-bytes** • **lpbk-drop-bps**
/components/component/subcomponents/subcomponent[name='FPC*ID*']/properties/property/[name=' *counter*']/state/value	Sensor for packet forwarding engine statistics. The subcomponent name **FPC*id*** refers to the number of the Flexible PIC Concentrator. This sensor provides visibility into packet forwarding engine errors and drops. This sensor pulls CPU counters. Supported on MX Series routers and PTX Series routers starting with Junos OS Release 17.4R1. The value for ***counter*** is one of the following; • **lts-hw-input-drops** • **lts-input-packets** • **lts-output-packets** • **lts-sw-input-control-drops** • **lts-sw-input-high-drops** • **lts-sw-input-medium-drops** • **lts-sw-input-low-drops** • **lts-sw-output-low-drops**
/components/component[name='CB0']/properties/property[name='*state*'/	Sensor for Control Board (CB) state information. This information can also be found using the operational mode command **show chassis hardware**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component[name='CB0']/properties/property[name='*manufacture-date*'/	Sensor for Control Board (CB) manufacturing date information. This information can also be found using the operational mode command **show chassis hardware extensive**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='PDU0:PSM0']/properties/property[name='*state*'/	Sensor for power distribution units (PDUs) state information. This information can also be found using the operational mode command **show chassis environment**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='PDU0:PSM0'/properties/property[name='*fru-model-number*'/	Sensor for field-replaceable unit (FRU) for a power distribution unit (PDU). This information can also be found using the operational mode command **show chassis hardware models**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='PDU0:PSM0'/properties/property[name='*manufacture-date*'/	Sensor for a power distribution unit (PDU) manufacturing date. This information can also be found using the operational mode command **show chassis hardware extensive**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='FPM Board']/properties/property[name='*state*'/	Sensor for state information for a craft interface (FPM). This information can also be found using the operational mode command **show chassis hardware models**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='FPM Board']/properties/property[name='*fru-model-number*'/	Sensor for field-replaceable unit (FRU) for a craft interface (FPM). This information can also be found using the operational mode command **show chassis hardware models**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='FPM Board']/properties/property[name='*manufacture-date*'/	Sensor for a craft interface (FPM) manufacturing date. This information can also be found using the operational mode command **show chassis hardware extensive**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='SIB0']/properties/property[name='*state*'/	Sensor for Switch Interface Boards (SIBs). **SIB0** and **SIB1** are supported. This information can also be found using the operational mode command **show chassis sibs**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component[name='FPC0']/properties/property	Sensor for the Flexible PIC Concentrator (FPC). This information can also be found using the operational mode command **show chassis fpc detail**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 are supported. You can also add the following as the end path: • **[name='*state*']** • **[name='*manufacture-date*']** • **[name='*uptime*']** • **[name='*Ambient Temp. EXHAUST*']** NOTE: This information can also be found using the operational mode command **show chassis environment fpc**. • **[name='*Ambient Temp. INLET*']** NOTE: This information can also be found using the operational mode command **show chassis environment fpc**. • **[name='*fru-model-number*']** NOTE: This information can also be found using the operational mode command **show chassis harware models**.
/components/component[name='FPC0:PIC0']/properties/property	Sensor for the physical interface card (PIC). This information can also be found using the operational mode command **show chassis pic fpc-slot *slot-num* pic-slot *slot-num***. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported. You can also add the following as the end path: • **[name='*state*']** • **[name='*uptime*']**
/components/component[name='Routing Engine 0']/properties/property[name='*state*']	Sensor for the routing engine state. This information can also be found using the operational mode command **show chassis environment routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*mastership-state*']	Sensor for the routing engine master status. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component[name='Routing Engine 0']/properties/property[name='*mastership-priority*'/	Sensor for the routing engine mastership election priority. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*Ambient Left*'/ /components/component[name='Routing Engine 0']/properties/property[name='*Ambient Right*'/	Sensor for the routing engine ambient temperature, both left and right. This information can also be found using the operational mode command **show chassis environment routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*firmware_rev*'/	Sensor for the routing engine's firmware revision. This information can also be found using the operational mode command **show chassis routing-engine bios**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*CPU Temperature*'/	Sensor for the routing engine's CPU temperature. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*memory-dram-used*'/ /components/component[name='Routing Engine 0']/properties/property[name='*memory-utilization-buffer*'/	Sensors for the routing engine's memory utilization. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*cpu-utilization-user*'/ /components/component[name='Routing Engine 0']/properties/property[name='*cpu-utilization-background*'/ /components/component[name='Routing Engine 0']/properties/property[name='*cpu-utilization-kernel*'/ /components/component[name='Routing Engine 0']/properties/property[name='*cpu-utilization-interruot*'/ /components/component[name='Routing Engine 0']/properties/property[name='*cpu-utilization-idle*'/	Sensors for the routing engine's CPU utilization. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component[name='Routing Engine 0']/properties/property[name='*uptime*'/	Sensor for routing engine **uptime**. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*reboot-reason*'/	Sensor for the cause of a routing engine reboot. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Routing Engine 0']/properties/property[name='*manufacture-date*'/	Sensor for the manufacture date of a routing engine. This information can also be found using the operational mode command **show chassis routing-engine**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Fan Tray0']/properties/property[name='*state*'/	Sensor for the fan tray. This information can also be found using the operational mode command **show chassis environment**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Fan Tray0']/properties/property[name='*fru-model-number*'/	Sensor for the fan tray model number. This information can also be found using the operational mode command **show chassis hardware models**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='Fan Tray0']/properties/property[name='*manufacture-date*'/	Sensor for the manuafacture date of the fan tray. This information can also be found using the operational mode command **show chassis hardware models**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component[name='PDU0:PSM0']/properties/property[name='*state*'/	Sensor for the Power Distribution Module (PDU) status. This information can also be found using the operational mode command **show chassis environment**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/components/component['Chassis']/properties/property[name='*state*'/	Sensor for the chassis state. The chassis state is always **ONLINE**. This information can also be found using the operational mode command **show chassis environment**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/components/component['PSM2']/properties/property[name='*fru-model-number*'/ **/components/component['PSM2']/properties/property[name='*manufacture-date*'/** **/components/component['PSM2']/properties/property[name='*Temperature*'/** **/components/component['PSM2']/properties/property[name='*state*'/**	Sensors for the power supply module (PSM) FRU model number, manufacture date, temperature, and state. This information can also be found using the operational mode command **show chassis environment**. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.
/junos/chassis/gres/	Sensor for graceful Routing Engine switchover (GRES) information. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported. You can also add the following as the end path for /junos/chassis/gres/: • **configured-state** • **error-state** • **gres-time** • **master-kernel-ready** • **slave-connect-time** • **slave-kernel-ready**
/junos/chassis/issu/	Sensor for in-service software upgrade (ISSU) information. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported. You can also add the following as the end path for /junos/chassis/issu/: • **failure-stage** • **current-issu-stage**
/junos/kernel/peer-infra/	Sensor for PFEMAN connection information. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported. You can also add the following as the end path for /junos/kernel/peer-infra/: • **pfeman-conn-drops** • **spurious-ppt-wkups**
/junos/kernel/record-meta-data/record_time	Sensor for system time at which Routing Engine metadata is created. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/kernel-ifstate	Sensor for Routing Engine ifstate information. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported. You can also add the following as the end path for /junos/kernel-ifstate/: • **alive-clients-cnt** • **alive-ifstates-cnt** • **client-limit-reached** • **dead-clients-cnt** • **dead-ifstates-cnt** • **delayed-unrefs-cnt** • **delayed-unrefs-max** • **stuck-clients-cnt**
/junos/kernel-ifstate/stats/churn-rate	Sensor for Routing Engine network object churn rate statistics. Starting in Junos OS Release 18.2R1, MX Series and PTX Series switches are supported. • **overall-churn-rate** • **route-add-rate** • **route-change-rate** • **route-delete-rate** • **nexthop-add-rate** • **nexthop-change-rate** • **nexthop-delete-rate**
/junos/kernel-ifstate/stats/peer-consumption-rate	Sensor for Routing Engine network object peer consumption rate statistics. Starting in Junos OS Release 18.2R1, MX Series and PTX Series switches are supported. • **peer-index** • **consumption-rate-counter** • **consumption-route-add-rate** • **consumption-route-delete-rate** • **consumption-nexthop-add-rate** • **consumption-nexthop-change-rate** • **consumption-nexthop-delete-rate**
/junos/kernel-ifstate/stats/record-seq-num	Sequence number of a statistic or record.
/junos/kernel-ifstate/stats/record-time	System time at which a statistic or record is created.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/kernel-ifstate/stats/vetos-statistics	Sensor for Routing Engine state statistics. Starting in Junos OS Release 18.2R1, MX Series and PTX Series switches are supported. • **veto-vm-page-count-severe** • **veto-ifstate-memory** • **veto-memory-overconsumed** • **veto-pfe-veto-max-routes** • **veto-too-many-delayed-unrefs** • **veto-nh-memory-usage** • **veto-mbuf-cluster** • **veto-flabel-space-exhaustion** • **veto-flabel-space-consumption**
/junos/ike-security-associations/ike-security-association/routing-instance [name=' *routing-instance-name*]	Sensor for Internet Key Exchange (IKE) security statistics. When you configure a subscription request, use the **reporting-interval** parameter to configure the interval (in seconds) in which statistics are reported. Starting with Junos OS Release 18.1R1, MX Series routers are supported. • **remote-ip** • **local-ip** • **number-ipsec-sa-created** • **number-ipsec-sa-deleted** • **number-ipsec-sa-rekey** • **exchange-type** • **in-bytes** • **in-packets** • **out-bytes** • **out-packets** • **delete-payload-received** • **delete-payload-transmitted** • **dpd-request-payload-received** • **dpd-request-payload-transmitted** • **dpd-response-payload-received** • **dpd-response-payload-transmitted** • **dpd-response-payload-missed** • **dpd-response-payload-maximum-delay** • **dpd-response-seq-payload-missed** • **invalid-spi-notify-received** • **invalid-spi-notify-transmitted** • **routing-instance**

Table 6: gRPC Sensors (continued)

resource path	Description
junos/rpm/probe-results/probe-test-results/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for probe test results for Real time Performance Monitoring (RPM) statistics. These statistics provide RPM monitoring data results collected by Juniper devices. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. The following end paths are also supported for the resource path: • owner • test-name • target-address • target-url • source-address • destination-interface • probe-type • icmp-id • routing-instance-name • test-size • http-status • loss-thresh-total • loss-thresh-succ • rtt-thresh • rtt-jitter-thresh • rtt-stddev-thresh • igr-thresh • igr-jitter-thresh • igr-stddev-thresh • egr-thresh • egr-jitter-thresh • egr-stddev-thresh • probe-tests-hw-ts-err/invalid-client-recv-ts-cntr • probe-tests-hw-ts-err/invalid-client-nots-cntr • probe-tests-hw-ts-err/invalid-server-send-ts-cntr • probe-tests-hw-ts-err/invalid-server-spent-time-cntr • probe-single-results • probe-single-results/probe-time • probe-single-results/probe-sent-time • probe-single-results/probe-status • probe-single-results/hardware-timestamp-status • probe-single-results/rtt • probe-single-results/egress • probe-single-results/ingress • probe-single-results/round-trip-jitter • probe-single-results/egress-jitter • probe-single-results/round-trip-interarrival-jitter

Table 6: gRPC Sensors (continued)

resource path	Description
	• probe-single-results/egress-interarrival-jitter • probe-single-results/ingress-interarrival-jitter • probe-test-generic-result • probe-test-generic-results/results-scope • probe-test-generic-results/probes-sent • probe-test-generic-results/probe-responses • probe-test-generic-results/loss-percentage • probe-test-generic-measurements • probe-test-generic-measurements/samples • probe-test-generic-measurements/min-delay • probe-test-generic-measurements/max-delay • probe-test-generic-measurements/avg-delay • probe-test-generic-measurements/jitter-delay • probe-test-generic-measurements/stddev-delay • probe-test-generic-measurements/sum-delay

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/rpm/history-results/history-single-test-results/	Sensor for history results for Real time Performance Monitoring (RPM) statistics. These statistics provide RPM monitoring data results collected by Juniper devices. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. The following end paths are also supported for the resource path: • **owner** • **test-name** • **target-address** • **target-url** • **source-address** • **destination-interface** • **probe-type** • **icmp-id** • **test-size** • **http-status** • **routing-instance-name** • **loss-thresh-total** • **loss-thresh-succ** • **rtt-thresh** • **rtt-jitter-thresh** • **rtt-stddev-thresh** • **igr-thresh** • **igr-jitter-thresh** • **igr-stddev-thresh** • **egr-thresh** • **egr-jitter-thresh** • **egr-stddev-thresh** • **probe-single-results** • **probe-single-results/probe-time** • **probe-single-results/probe-sent-time** • **probe-single-results/probe-status** • **probe-single-results/hardware-timestamp-status** • **probe-single-results/rtt** • **probe-single-results/egress** • **probe-single-results/ingress** • **probe-single-results/round-trip-jitter** • **probe-single-results/egress-jitter** • **probe-single-results/ingress-jitter** • **probe-single-results/round-trip-interarrival-jitter** • **probe-single-results/egress-interarrival-jitter** • **probe-single-results/ingress-interarrival-jitter**

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/rpm/server/	Sensor for server results for Real time Performance Monitoring (RPM) statistics. These statistics provide RPM monitoring data results collected by Juniper devices. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. The following end paths are also supported for the resource path: • **active-servers** • **active-servers/protocol** • **active-servers/port** • **active-servers/dst-interface**
/junos/security/veriexec-state	Sensor for Veriexec state information. Starting with Junos OS Release 19.1R1, EX9200, EX9251, EX9253, MX Series, and PTX Series are supported.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/services/label-switched-path/usage/	Sensor for LSP statistics. On MX Series routers only, the following are also supported: bidirectional LSPs for ultimate-hop popping (UHP). Starting with Junos OS Release 17.2R1, QFX10000 switches and PTX1000 routers are also supported. Starting with Junos OS Release 17.3R1, EX9200 switches are also supported. 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. Starting with Junos OS Release 18.2R1, QFX5100, QFX5110, and QFX5200 switches are also supported. . Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. Starting with Junos OS Release 19.1R1, PTX10001-20C routers are supported for RSVP LSPs only. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported. For bypass LSPs, the following are exported: • 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. When the bypass LSP is active, traffic is exported both on the bypass LSP and the ingress (protected) LSP. 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.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/twamp/client/control-connection/	Sensor for client control connection results for Two-Way Active Management Protocol (TWAMP). TWAMP (described in RFC 5357). Used to measure traffic performance between end-points, you can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. The following end paths are also supported for the resource path: • **control-name** • **client-address** • **client-port** • **server-address** • **server-port** • **session-count** • **auth-mode** • **server-address** • **server-port** • **test-session/session-name** • **test-session/sender-address** • **test-session/sender-port** • **test-session/reflector-address** • **test-session/reflector-port**

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/twamp/client/probe-test-results/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for client probe test results for Two-Way Active Management Protocol (TWAMP). TWAMP (described in RFC 5357) is used to measure traffic performance between end-points. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. The following end paths are also supported for the resource path: • owner • test-name • destination-interface • test-size • server-address • server-port • client-address • client-port • reflector-address • reflector-port • sender-address • sender-port • loss-thresh-total • loss-thresh-succ • rtt-thresh • rtt-jitter-thresh • rtt-stddev-thresh • igr-thresh • igr-jitter-thresh • igr-stddev-thresh • egr-thresh • egr-jitter-thresh • egr-stddev-thresh • probe-tests-hw-ts-err/invalid-client-recv-ts-cntr • probe-tests-hw-ts-err/invalid-client-nots-cntr • probe-tests-hw-ts-err/invalid-server-send-ts-cntr • probe-tests-hw-ts-err/invalid-server-spent-time-cntr • probe-single-results/ • probe-single-results/probe-time • probe-single-results/probe-sent-time • probe-single-results/probe-status • probe-single-results/hardware-timestamp-status • probe-single-results/rtt • probe-single-results/egress • probe-single-results/ingress • probe-single-results/round-trip-jitter

Table 6: gRPC Sensors (continued)

resource path	Description
	• probe-single-results/egress-jitter • probe-single-results/ingress-jitter • probe-single-results/round-trip-interarrival-jitter • probe-single-results/egress-interarrival-jitter • probe-single-results/ingress-interarrival-jitter • probe-test-generic-results/ • probe-test-generic-results/results-scope • probe-test-generic-results/probes-sent • probe-test-generic-results/probe-responses • probe-test-generic-results/loss-percentage • probe-test-generic-results/probe-test-rtt • probe-test-generic-results/probe-test-generic-measurements/ • probe-test-generic-results/probe-test-generic-measurements//probe-measurement-type • probe-test-generic-results/probe-test-generic-measurements/samples • probe-test-generic-results/probe-test-generic-measurements/min-delay • probe-test-generic-results/probe-test-generic-measurements/max-delay • probe-test-generic-results/probe-test-generic-measurements/avg-delay • probe-test-generic-results/probe-test-generic-measurements/jitter-delay • probe-test-generic-results/probe-test-generic-measurements/stddev-delay • probe-test-generic-results/probe-test-generic-measurements/sum-delay

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/twamp/client/history-test-results/ history-single-test-results/	Sensor for client history test results for Two-Way Active Management Protocol (TWAMP). TWAMP (described in RFC 5357) is used to measure traffic performance between end-points. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. • **owner** • **test-name** • **destination-interface** • **test-size** • **server-address** • **server-port** • **client-address** • **client-port** • **reflector-address** • **reflector-port** • **sender-address** • **sender-port** • **loss-thresh-total** • **loss-thresh-succ** • **rtt-thresh** • **rtt-jitter-thresh** • **rtt-stddev-thresh** • **igr-thresh** • **igr-jitter-thresh** • **igr-stddev-thresh** • **egr-thresh** • **egr-jitter-thresh** • **egr-stddev-thresh** • **probe-single-results/** • **probe-single-results/probe-time** • **probe-single-results/probe-sent-time** • **probe-single-results/probe-status** • **probe-single-results/hardware-timestamp-status** • **probe-single-results/rtt** • **probe-single-results/egress** • **probe-single-results/ingress** • **probe-single-results/round-trip-jitter** • **probe-single-results/egress-jitter** • **probe-single-results/ingress-jitter** • **probe-single-results/round-trip-interarrival-jitter** • **probe-single-results/egress-interarrival-jitter** • **probe-single-results/ingress-interarrival-jitter**

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/twamp/server/control-connection/	Sensor for control connection results for servers for Two-Way Active Management Protocol (TWAMP). TWAMP (described in RFC 5357) is used to measure traffic performance between end-points. You can use this information to assure service level agreements, improve network design, and optimize traffic engineering. Starting with Junos OS Release 18.3R1, MX Series routers are supported. • **control-name** • **client-address** • **client-port** • **server-address** • **server-port** • **session-count** • **auth-mode** • **test-session/** • **test-session/session-name** • **test-session/sender-address** • **test-session/sender-port** • **test-session/reflector-address** • **test-session/reflector-port**

Table 6: gRPC Sensors (continued)

resource path	Description
/network-instances/ network-instance[name='*instance-name*']/mpls/lsps/ constrained-path/container-tunnels/ container-tunnel[name='*name*']state/	Sensor for container tunnel streaming notifications and normalization status. Starting with Junos OS Release 19.1R1, this sensor is supported on all platforms supporting JTI. The following paths are also supported: • **name** • **oper-state** • **member-count** • **minimum-lsp-count** • **maximum-lsp-count** • **normalize-timer** • **normalize-threshold** • **aggregate-bandwidth** • **sampled-aggregate-bandwidth** • **max-signaling-bandwidth** • **min-signaling-bandwidth** • **splitting-bandwidth** • **merging-bandwidth** • **incremental-normalization** • **failover-normalization** • **time-to-normalize** • **sampling/sampling-outlier-cutoff** • **sampling/sampling-mode** • **sampling/sampling-percentile** • **normalization-status** which includes **normalize-init**, **normalize-complete**, and **avoid-normalize**. normalization-status/normalize-init normalization-status/normalize-complete normalization-status/avoid-normalize

Table 6: gRPC Sensors (continued)

resource path	Description
/network-instances/ network-instance[name='*instance-name*']/mpls/lsps/ constrained-path/tunnels/tunnel[name='*name*']/ p2p-tunnel-attributes/p2p-primary-paths/ p2p-primary-path[name='*path-name*']/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for LSP events and properties. LSP events and properties are exported for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs. 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 **/network-instances/network-instance[name='*instance-name*']/** in front of all supported paths. . Supported on MX Series and PTX Series routers and QFX10000 switches starting with Junos OS Release 17.2R1. Supported on EX4600 and EX9200 switches and QFX5110 and QFX5200 switches starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. The following paths are also supported: • **lsp-instances/state/notify-status/initiated** • **lsp-instances/state/notify-status/lsp-up** • **lsp-instances/state/notify-status/lsp-down** • **lsp-instances/state/notify-status/lp-available** • **lsp-instances/state/notify-status/lp-unavailable** • **lsp-instances/state/notify-status/autobw-success** • **lsp-instances/state/notify-status/autobw-fail** • **lsp-instances/state/notify-status/patherr-recv** • **lsp-instances/state/notify-status/tunnel-local-repaired** • **lsp-instances/state/bandwidth** • **lsp-instances/state/metric** • **lsp-instances/state/max-avg-bandwidth** • **/state/associated-rsvp-sessions/associated-rsvp-session[local-index ='*index-number*']/** • **state/notify-status** • **state/notify-status/select-active-path** • **state/notify-status/deselect-active-path** • **state/notify-status/change-active-path** • **state/notify-status/originate-mbb** • **state/notify-status/cspf-noroute** • **state/notify-status/cspf-success** • **state/notify-status/gr-recovery-fail** • **state/explicit-path-name** NOTE: To specify a specific LSP name and source address, include

Table 6: gRPC Sensors (continued)

resource path	Description
	[name='lsp-name',source='address'] after **mpls/lsp/constrained-path-tunnels/tunnel/** in any of the supported paths. If do not include a specific LSP name, data is exported for all configured LSPs.
/network-instances/ network-instance[name='*instance-name*']/mpls/lsps/ constrained-path/tunnels/tunnel[name='*name*']/ p2p-tunnel-attributes/p2p-primary-paths/ p2p-primary-path[name='*path-name*'][local-index='local-index']/ state/notify-status	Sensor for self-ping failure. This sensor supports self-ping logs. Starting with Junos OS Release 19.1R1, this sensor is supported on all platforms supporting JTI.
/network-instances/ network-instance[name='*instance-name*']/mpls/lsps/ constrained-path/tunnels/tunnel[name='*name*']/ p2p-tunnel-attributes/p2p-primary-paths/ p2p-primary-path[name='*path-name*'][local-index='local-index']/ state/reason/	Sensor that indicates the reason for a self-ping failure. Starting with Junos OS Release 19.1R1, this sensor is supported on all platforms supporting JTI.
/network-instances/ network-instance[name='*instance-name*']/mpls/ signaling-protocols/rsvp-te/sessions/session/state/ notify-status	Starting with Junos OS Release 17.4R1, telemetry data for LSP events and properties is reported separately for each routing instance. Supported on MX Series and PTX Series routers and QFX10000 switches starting with Junos OS Release 17.2R1. Supported on EX4600 and EX9200 switches and QFX5110 and QFX5200 switches starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. The following paths are also supported: • **detour-up** • **detour-down** • **patherr-recv** • **patherr-recv/admission_control_failure** • **patherr-recv/session_preeempted** • **patherr-recv/bad_loose_route** • **patherr-recv/bad_strict_route** • **patherr-recv/label_allocation_failure** • **patherr-recv/non_rsvp_capable_router** • **patherr-recv/ttl_expired** • **patherr-recv/routing_loop_detected** • **patherr-recv/requested_bandwidth_unavailable** • **patherr-recv/ttl_expired** • **pathmtu-change**

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/npu-memory/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for network processing unit (NPU) memory, NPU memory utilization, and total memory available for each memory type. Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1. Supported on EX9200 switches starting with Junos OS Release 17.3R1. NOTE: Starting with Junos Release 17.4R1, FPC1 and FCP2 on PTX Series routers export data for NPU memory and NPU memory utilization. Previously, this sensor was supported only on FPC 3. Starting with Junos OS Release 18.3R1, EX4650 switches are supported. Starting with Junos OS Release 19.1R1, PTX10002 routers are supported. The OpenConfig path is **/components/component[name="FPC<*fpc-id*>:NPU<*npu-id*>"] /properties/property/** You can also add the following to the end of the path to stream specific statistics for NPU memory: • **[name="mem-util-<*memory-name*>-size"]/value** • **[name="mem-util-<*memory-name*>-bytes-allocated"]/value** • **[name="mem-util-<*memory-name*>-utilization"]/value** • **[name="mem-util-<*partition-name*>-< *app-name*>-allocation-count"]/value** • **[name="mem-util-<*partition-name*>-< *app-name*>-bytes-allocated"]/value** • **[name="mem-util-<*partition-name*>-< *app-name*>-free-count"]/value** You can add the following to the end of the path to stream specific statistics for NPU utilization: • **[name="util-<*memory-name*>-average-util">/value** • **[name="util-<*memory-name*>-highest-util">/value** • **[name="util-<*memory-name*>-lowest-util">/value** • **[name="util-<*memory-name*>-average-cache-hit-rate">/value** • **[name="util-<*memory-name*>-lowest-cache-hit-rate">/value** • **[name="util-<*packet-identifier*>-rate">/value** You can also export the following statistics for NPU memory for PTX routers only • **pfe_name** • **combined_pool_name** • **combined_size** • **combined_usage_cnt** • **combined_utilization** • **global_pool_name** • **global_usage_cnt** • **global_alloc_cnt** • **global_free_cnt**

Table 6: gRPC Sensors (continued)

resource path	Description
	• **local_pool_name** • **local_usage_cnt** • **local_alloc_cnt** • **local_free_cnt**
/junos/system/linecard/node-slicing/af-fab-stats/	Sensor to export abstracted fabric (AF) interface-specific load-balancing and fabric queue statistics. This sensor is only supported for in node virtualization configurations on MX routers with an AF Interface as the connecting link between guest network functions (GNFs).The sensor also reports aggregated statistics across all AF interfaces hosted on a source packet forwarding engine of local guest GNFs along with the fabric statistics for all traffic ingressing from and egressing to the fabric from that the packet forwarding engine. Supported on MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers with Junos OS Release 18.4R1.
/junos/system/linecard/cpu/memory/	Sensor for CPU memory. NOTE: On PTX Series routers, FPC1 and FPC2 are not supported. Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1. Supported on EX9200 switches starting with Junos OS Release 17.3R1. Supported on QFX5100, QFX5110, and QFX5200 switches starting with Junos OS Release 18.2R1. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. You can also include the following to end of the resource path for CPU memory: • **[name="mem-util-<*memory-name*>-size"]/value** • **[name="mem-util-<*memory-name*>-bytes-allocated"]/value** • **[name="mem-util-<*memory-name*>-utilization"]/value** • **[name="mem-util-<*memory-name*>-< *app-name*>-allocations"]/value** • **[name="mem-util-<*memory-name*>-< *app-name*>-frees"]/value** • **[name="mem-util-<*memory-name*>-<*app-name*>-allocations-failed"]/value**
/junos/system/linecard/npu/utilization/	Packet Forwarding Engine sensor for NPU processor utilization. Junos OS Release 19.1R1 and later on PTX10002 routers.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/system/linecard/interface/	Packet Forwarding Engine sensor for physical interface traffic. 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. For Aggregated Ethernet interfaces, statistics are exported for the member physical interfaces. You must aggregate the counters at the destination server, or collector. If a physical interface is administratively down or operationally down, interface counters are not exported. Issuing an operational **clear** command, such as **clear interfaces statistics all**, does not reset statistics exported by the line card. Supported on PTX Series routers starting with Junos OS Release 15.1F3. Supported on MX Series routers starting with Junos OS Release 15.1F5. Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1. Supported on EX9200 switches, QFX5110 switches, and MX150 routers starting with Junos OS Release 17.3R1. Supported on QFX5100, QFX5110, and QFX5200 switches starting with Junos OS Release 18.2R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches Starting with Junos OS Release 18.4R1. Supported on QFX10002 switches and PTX10002 routers starting with Junos OS Release 19.1R1.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/system/linecard/interface/logical/usage	Packet Forwarding Engine sensor for logical interface statistics. NOTE: If a logical interface is operationally down, interface statistics continue to be exported. Issuing an operational **clear** command, such as **clear interfaces statistics all**, does not reset statistics exported by the line card. NOTE: Locally injected packets from the Routing Engine are not exported. Supported in Junos OS Release 15.1F5. Supported QFX10000 switches starting with on Junos OS Release 17.2R1. Supported on EX9200 switches, QFX5110 switches, and MX150 routers starting with Junos OS Release 17.3R1. Supported on QFX5100, QFX5110, and QFX5200 switches starting with Junos OS Release 18.2R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1.
/junos/system/linecard/optical	Sensor for optical alarms. Configure this sensor for **et-*type-fpc/pic/port*** (100-Gigabit Ethernet) interfaces. Supported on ACX6360 Universal Metro, MX Series, and PTX Series routers with a CFP2-DCO optics module starting with Junos OS Release 18.3R1. This module provides a high-density, long-haul OTN transport solution with MACSec capability.
/junos/system/linecard/otn	Sensor for G.709 optical transport network (OTN) alarms. Configure this sensor on **ot-*type-fpc/pic/port*** interfaces. Supported on ACX6360 Universal Metro, MX Series, and PTX Series routers with a CFP2-DCO optics module starting with Junos OS Release 18.3R1. This module provides a high-density, long-haul OTN transport solution with MACSec capability.
/junos/system/linecard/qmon-sw/	Sensor for congestion and latency monitoring statistics. Supported on QFX5100, QFX5110, and QFX5200 switches starting with Junos OS Release 18.2R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1.

Table 6: gRPC Sensors (continued)

resource path	Description
/network-instances/network-instance/protocols/protocol/bgp/ 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. For BGP routing table paths, the **/network-instances/network-instance/** path is not supported. Each address family supports exporting data for five different tables, a main routing table, and four per-neighbor tables: • local-rib— main BGP routing table for the main routing instance. • adj-rib-in-pre— NLRI updates received from the neighbor before any local input policy filters have been applied. • adj-rib-in-post— routes received from the neighbor eligible for best-path selection after local input policy filters have been applied. • adj-rib-out-pre— routes eligible for advertising to the neighbor before output policy filters have been applied. • adj-rib-out-post— routes eligible for advertising to the neighbor after output policy filters have been applied. 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: • **/bgp-rib/afi-safis/afi-safi/ipv4-unicast/loc-rib/** • **/bgp-rib/afi-safis/afi-safi/ipv6-unicast/loc-rib/** • **/bgp-rib/afi-safis/afi-safi/ipv4-unicast/neighbors/neighbor/adj-rib-in-pre/** • **/bgp-rib/afi-safis/afi-safi/ipv6-unicast/neighbors/neighbor/adj-rib-in-pre/** • **/bgp-rib/afi-safis/afi-safi/ipv4-unicast/neighbors/neighbor/adj-rib-in-post/** • **/bgp-rib/afi-safis/afi-safi/ipv6-unicast/neighbors/neighbor/adj-rib-in-post/** • **/bgp-rib/afi-safis/afi-safi/ipv4-unicast/neighbors/neighbor/adj-rib-out-pre/** • **/bgp-rib/afi-safis/afi-safi/ipv6-unicast/neighbors/neighbor/adj-rib-out-pre/** • **/bgp-rib/afi-safis/afi-safi/ipv4-unicast/neighbors/neighbor/adj-rib-out-post/** • **/bgp-rib/afi-safis/afi-safi/ipv6-unicast/neighbors/neighbor/adj-rib-out-post/**	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for BGP peer information. Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers, EX4600 and EX9200 switches, and QFX5110 switches starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. NOTE: Starting with Junos OS Release 17.3R1, telemetry data streamed through gRPC for BGP peers is reported separately for each configured routing instance. 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: **/network-instances/network-instance[name_'instance-name']/protocols/protocol/** Starting with Junos OS Release 17.3R1, the following paths are also supported: • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/prefixes/accepted** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/prefixes/rejected** **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/active** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/queues/output** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-safis/afi-safi/state/queues/input** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/snmp-peer-index** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/ImportEval** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/ImportEvalPending** • **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/messages/received/notification** **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/messages/sent/notification** **/network-instances/network-instance/protocols/protocol/bgp/transport/state/remote-port** **/network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/state/supported-capabilities** 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: **/network-instances/network-instance[name_'instance-name']/protocols/protocol/**

Table 6: gRPC Sensors (continued)

resource path	Description
	You can also include the following at the end path to **/network-instances/network-instance[name_'instance-name']/protocols/protocol/bgp/neighbors/neighbor/**: • **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** You can also include the following at the end path to **/network-instances/network-instance[name_'instance-name']/protocols/protocol//bgp/global/**: • **afi-safis/afi-safi/state/total-prefixes** You can also include the following at the end path to **/network-instances/network-instance[name_'instance-name']/protocols/protocol//bgp/peer-groups/peer-group[name_'peer-group-name]/**: • **afi-safis/afi-safi/add-paths/eligible-prefix-policy** • **state/peer-count/** NOTE: For paths that export data for BGP routing tables, which are supported starting with Junos OS Release 17.4R1, you can apend the following to each of the paths:

Table 6: gRPC Sensors (continued)

resource path	Description
	• /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 6: gRPC Sensors (continued)

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

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for memory utilization for routing protocol task. Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers, EX4600 and EX9200 switches and QFX5110 switches starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. You can also include the following at the end path to **/junos/task-memory-information/**: • 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

Table 6: gRPC Sensors (continued)

resource path	Description
	• task-memory-max-page-data-bytes • task-memory-dir-bytes • task-memory-max-dir-bytes • task-memory-total-bytes-in-use • task-memory-total-bytes-percent

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/system/linecard/firewall/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for firewall filter counters and policer counters. Each line card reports counters separately. Supported on QFX10000 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1. Supported on QFX5100, QFX5110, and QFX5200 switches starting with Junos OS Release 18.2R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported. 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. Firewall counters are exported even if the interface to which the firewall filter is attached is operationally down. The following OpenConfig paths are supported: • 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='*couner-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

Table 6: gRPC Sensors (continued)

resource path	Description
	(MX Series only) • **/junos/firewall/firewall-stats/[name='*filter-name*']/hierarchical-policer-stats/[name='*hierarchical-policer-name*']/aggregate-bytes** (MX Series only)

Table 6: gRPC Sensors (continued)

resource path	Description
/interfaces/interface/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for physical interface traffic. 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. For Aggregated Ethernet interfaces, statistics are exported for the member physical interfaces. You must aggregate the counters at the destination server, or collector. If a physical interface is administratively down or operationally down, interface counters are not exported. Only fields with a non-zero value are exported. Supported on QFX10000 switches and PTX1000 routers starting with Junos OS Release 17.2R1. Supported on EX9200 switches and MX150 routers starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported. Starting with Junos OS Release 18.3R1, when a subscription is made to **/interfaces** on MX, EX, QFX, PTX, and ACX platforms, traffic and queue statistics are delivered in two separate sensors. This can reduce the reap time for non-queue data for platforms supporting Virtual Output Queues (VOQ), such as PTX Series routers. The two sensors are: • **/junos/system/linecard/interface/traffic/** exports all fields except queue statistics. • **/junos/system/linecard/interface/queue/** exports queue statistics. NOTE: End paths supporting ON_CHANGE streaming are indicated. The following paths are also supported: • **/interfaces/interface[name='*interface-name*']/state/parent_ae_name** • **/interfaces/interface[name='*interface-name*']/state/admin-status** ON_CHANGE streaming supported for Junos Os and for Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/state/counters/carrier-transitions** • **/interfaces/interface[name='*interface-name*']/state/last-change** • **/interfaces/interface[name='*interface-name*']/state/high-speed** • **/interfaces/interface[name='*interface-name*']/state/counters/out-octets** • **/interfaces/interface[name='*interface-name*']/state/counters/out-unicast-pkts**

Table 6: gRPC Sensors (continued)

resource path	Description
	• /interfaces/interface[name='*interface-name*']/state/counters/out-multicast-pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-broadcast-pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-errors • /interfaces/interface[name='*interface-name*']/state/counters/in-octets • /interfaces/interface[name='*interface-name*']/state/counters/in-unicast-pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-multicast-pkts • /interfaces/interface[name='*interface-name*']/state/ • /interfaces/interface[name='*interface-name*']/state/counters/in-broadcast-pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-errors • /interfaces/interface[name='*interface-name*']/state/in-pause-pkts • /interfaces/interface[name='*interface-name*']/state/out-pause-pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/ • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/ pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/bytes • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/tail-drop-pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/ rl-drop-pkts • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/ rl-drop-bytes • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/avg-buffer-occupancy • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/cur-buffer-occupancy • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/peak-buffer-occupancy • /interfaces/interface[name='*interface-name*']/state/counters/in-queue [queue-number=*queue_number*]/allocated-buffer-size • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/bytes • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/tail-drop-pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/rl-drop-pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/ rl-drop-bytes • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/red-drop-pkts • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/red-drop-bytes • /interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/avg-buffer-occupancy

Table 6: gRPC Sensors (continued)

resource path	Description
	• **/interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/cur-buffer-occupancy** • **/interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/ peak-buffer-occupancy** • **/interfaces/interface[name='*interface-name*']/state/counters/out-queue [queue-number=*queue_number*]/allocated-buffer-size** • **/interfaces/interface[name='*interface-name*']/state/description** ON_CHANGE streaming supported for Junos OS and for Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/state/enabled** • **/interfaces/interface[name='*interface-name*']/state/ifindex** ON_CHANGE streaming supported for Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/state/last-change** • **/interfaces/interface[name='*interface-name*']/state/mtu** • **/interfaces/interface[name='*interface-name*']/state/name** • **/interfaces/interface[name='*interface-name*']/state/oper-status** ON_CHANGE streaming supported for Junos OS and for Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/state/type**

Table 6: gRPC Sensors (continued)

resource path	Description
/interfaces/interface/subinterfaces/ /interfaces/ interface[name='*interface-name*']/subinterfaces/ subinterface[index='*unit*']	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for logical interface traffic. NOTE: If a logical interface is operationally down, interface statistics continue to be exported. NOTE: Locally injected packets from the Routing Engine are not exported. Supported on QFX10000 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1. Starting with Junos OS Release 18.1R1, QFX5100 switches are also supported. Starting with Junos OS Release 18.3R1, QFX5120-48Y and EX4650 switches are also supported. Starting with Junos OS Release 18.4R1, EX4600 switches are also supported. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported. NOTE: End paths supporting ON_CHANGE streaming are indicated. The following paths are also supported: • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/name** ON_CHANGE streaming supported. This value does not change with an event, but will be streamed on event creation and deletion. • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/ifindex** ON_CHANGE streaming supported. This value does not change with an event, but will be streamed on event creation and deletion. • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/index** ON_CHANGE streaming supported. This value does not change with an event, but will be streamed on event creation and deletion. • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/snmp_index** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/admin_status** ON_CHANGE streaming supported for Junos OS and Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oper_status** ON_CHANGE streaming supported for Junos OS and Junos OS Evolved Release 19.1R1 and higher • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/last_change** • **/interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/description**

Table 6: gRPC Sensors (continued)

resource path	Description
	ON_CHANGE streaming supported for Junos OS and Junos OS Evolved Release 19.1R1 and higher • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/enabled • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_octets • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_unicast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_broadcast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_multicast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_discards • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_errors • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/in_unknown_protos • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_octets • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_unicast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_broadcast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_multicast_pkts • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_discards • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/out_errors • /interfaces/interface[name='*interface-name*']/subinterfaces/subinterface[index='*unit*']state/oounters/last_clear
/junos/system/linecard/optics/	Sensor for various optical interface performance metrics, such as transmit and receive power levels. Supported on QFX10000 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1. Supported on EX4650 switches starting with Junos OS Release 18.3R1.

Table 6: gRPC Sensors (continued)

resource path	Description
/junos/rsvp-interface-information/	Sensor for events and properties for RSVP interfaces. 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. Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers, QFX5110 switches, and EX4600 and EX9200 switches starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. You can also add the following to the end path for **/junos/rsvp-interace-information/**: • **active-count** • **rsvp-interface/interface-name** • **rsvp-interface/index** • **rsvp-interface/rsvp-status** • **rsvp-interface/authentication-flag** • **rsvp-interface/aggregate-flag** • **rsvp-interface/ack-flag** • **rsvp-interface/protect-flag** • **rsvp-interface/hello-interval** • **rsvp-interface/interface-address** • **message-statistics/rsvp-message** • **rsvp-interface/message-statistics/messages-sent** • **rsvp-interface/message-statistics/messages-received** • **rsvp-interrface/message-statistics/messages-sent-5seconds** • **rsvp-interface/message-statistics/messages-received-5seconds** • **rsvp-interface/rsvp-telink/active-reservation** • **rsvp-interface/rsvp-telink/preemption-count** • **rsvp-interface/rsvp-telink/update-threshold** • **rsvp-interface/rsvp-telink/subscription** • **rsvp-interface/rsvp-telink/static-bandwidth** • **rsvp-interface/rsvp-telink/available-bandwidth** • **rsvp-interface/rsvp-telink/reserved-bandwidth/bandwidth-priority** • **rsvp-interface/rsvp-telink/reserved-bandwidth/total-reserved-bandwidth**

Table 6: gRPC Sensors (continued)

resource path	Description
/components/	

Table 6: gRPC Sensors (continued)

resource path	Description
	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. NOTE: Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on EX9200 switches and MX150 routers starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. You can also add the following to each of the paths: • **name** • **cidx** • **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** The following paths are also supported only for Routing Engine statistics: • **properties/property/mastership-state** • **properties/property/mastership-priority**

Table 6: gRPC Sensors (continued)

resource path	Description
	• properties/property/temperature-cpu • properties/property/memory-dram-installed • properties/property/cpu-utilization-user • properties/property/cpu-utilization-background • properties/property/cpu-utilization-kernel • properties/property/cpu-utilization-idle • properties/property/reboot-reason The following paths are also supported for power modules: • properties/property/power-zone-upper-capacity • properties/property/power-zone-upper-maximum • properties/property/power-zone-upper-allocated • properties/property/power-zone-upper-remaining • properties/property/power-zone-upper-usage • properties/property/power-zone-lower-capacity • properties/property/power-zone-lower-maximum • properties/property/power-zone-lower-allocated • properties/property/power-zone-lower-remaining • properties/property/power-zone-lower-usage • properties/property/power-zone-0-capacity • properties/property/power-zone-0-maximum • properties/property/power-zone-0-allocated • properties/property/power-zone-0-remaining • properties/property/power-zone-0-usage • properties/property/power-zone-1-capacity • properties/property/power-zone-1-maximum • properties/property/power-zone-1-allocated • properties/property/power-zone-1-remaining • properties/property/power-zone-1-usage • properties/property/power-system-capacity • properties/property/power-system-allocated • properties/property/power-system-remaining • properties/property/power-system-usage • properties/property/temperature-ambient The following paths are supported for either Switch Fabric Board or Control Boards or both:

Table 6: gRPC Sensors (continued)

resource path	Description
	• **properties/property/temperature-zone-0-intake** (SFB only) • **properties/property/temperature-zone-0-intake-a** (both SFB and CB) • **properties/property/temperature-zone-1-intake-b** (both SFB and CB) • **properties/property/temperature-zone-0-exhaust** (SFB only) • **properties/property/temperature-zone-1-exhaust** (SFB only) • **properties/property/temperature-zone-0-intake-c** (CB only) • **properties/property/temperature-zone-0-exhaust-a** (CB only) • **properties/property/temperature-zone-1-exhaust-b** (CB only)

Table 6: gRPC Sensors (continued)

resource path	Description
/lacp/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for operational state of aggregated Ethernet interfaces configured with the Link Aggregation Control Protocol. Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers and EX9200 switches starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with unos OS Release 18.4R1, Supported on PTX10003 routers starting with Junos OS Evolved Release 19.1R1. You can also add the following to the end of the path for **/lacp/**: • **interfaces/interface/state** • **interfaces/interface/members/member/state/activity** • **interfaces/interface/members/member/state/timeout** • **interfaces/interface/members/member/state/system-id** • **interfaces/interface/members/member/state/partner-id** • **interfaces/interface/members/member/state/interface** • **interfaces/interface/members/member/state/synchronization** • **interfaces/interface/members/member/state/aggregatable** • **interfaces/interface/members/member/state/collecting** • **interfaces/interface/members/member/state/distributing** • **interfaces/interface/members/member/state/oper-key** • **interfaces/interface/members/member/state/partner-key** • **interfaces/interface/members/member/state/counters/lacp-in-packets** • **interfaces/interface/members/member/state/counters/lacp-out-packets** • **interfaces/interface/members/member/state/counters/lacp-rx-errors** • **interfaces/interface/members/member/state/counters/lacp-unknown-errors** • **interfaces/interface/members/member/state/counters/lacp-errors** • **state/system-priority** • **interfaces/interface[name='aggregate-interface-name']/state/** • **interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/** • **interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/counters/** • **interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/port-num** • **interfaces/interface[name='aggregate-interface-name']/members/member[interface='interface-name']/state/partner-port-num** • **interfaces/interface[name='aggregate-interface-name']**

Table 6: gRPC Sensors (continued)

resource path	Description
	/members/member[interface='interface-name']/state/mux-state

Table 6: gRPC Sensors (continued)

resource path	Description
/lldp/	

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for operational state of Ethernet interfaces enabled with the Link Layer Discovery Protocol. Supported on QFX10000 switches and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers and EX9200, EX4600, and QFX5110 switches starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. Supported on PTX10003 routers starting with Junos OS Evolved Release 19.1R1. ON_CHANGE streaming is supported on MX Series and PTX Series routers, starting with Junos OS Release 18.3R1. You can also add the following to the end of the path for **/lldp/**: NOTE: End paths supporting ON_CHANGE streaming are indicated. • **state/** • **state/enabled/** ON_CHANGE streaming supported • **state/hello-timer/** ON_CHANGE streaming supported • **state/chassis-id/** ON_CHANGE streaming supported • **state/chassis-id-type/** ON_CHANGE streaming supported • **state/system-name/** ON_CHANGE streaming supported • **state/system-description/** ON_CHANGE streaming supported • **state/loc-port-id-type/** • **state/counters/frame-in/** • **state/counters/frame-out/** • **state/counters/frame-error-in/** • **state/counters/frame-discard/** • **state/counters/tlv-unknown/** • **state/counters/tlv-discard/** • **state/counters/tlv-accepted/** • **state/counters/entries-aged-out/** • **state/counters/last-clear/** • **interfaces/interface[name='interface-name']/**

Table 6: gRPC Sensors (continued)

resource path	Description
	• interfaces/interface[name='interface-name']/state/ • interfaces/interface[name='interface-name']/state/name/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/state/enabled/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/state/loc-port-id/ • interfaces/interface[name='interface-name']/state/loc-port-description/ • interfaces/interface[name='interface-name']/state/counters/frame-in/ • interfaces/interface[name='interface-name']/state/counters/frame-error-in/ • interfaces/interface[name='interface-name']/state/counters/frame-discard/ • interfaces/interface[name='interface-name']/state/counters/tlv-discard/ • interfaces/interface[name='interface-name']/state/counters/tlv-unknown/ • interfaces/interface[name='interface-name']/state/counters/frame-out/ • interfaces/interface[name='interface-name']/state/counters/frame-error-out/ • interfaces/interface[name='interface-name']/state/counters/last-clear/ • interfaces/interface[name='interface-name']/neighbors/neighbor/ • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/cabapility/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/cabapility/name/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/cabapility/state/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/cabapility/state/name/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/capabilities/cabapility/state/enabled/ ON_CHANGE streaming supported • interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/ • interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/ • interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/type/ • interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/oui/

Table 6: gRPC Sensors (continued)

resource path	Description
	• **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/oui-subtype/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/state/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/state/type/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/state/oui/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/state/oui-subtype/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/custom-tlvs/tlv/state/value/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/system-name/** ON_CHANGE streaming supported. This resource path does not change with an event, but will be streamed on creation and deletion. • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/system-description/** ON_CHANGE streaming supported • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/chassis-id/** ON_CHANGE streaming supported. This resource path does not change with an event, but will be streamed on creation and deletion. • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/chassis-id-type/** ON_CHANGE streaming supported. This resource path does not change with an event, but will be streamed on creation and deletion. • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/id/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/age/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/last-update/** • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/port-id/** ON_CHANGE streaming supported • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/port-id-type/** ON_CHANGE streaming supported • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/port-description/** ON_CHANGE streaming supported • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/management-address/** ON_CHANGE streaming supported • **interfaces/interface[name='interface-name']/neighbors/neighbor/state/management-address-type/** ON_CHANGE streaming supported

Table 6: gRPC Sensors (continued)

resource path	Description

/mpls/lsps/constrained-path/tunnels/tunnel[name='*foo-name*',source='*foo-source*']/p2p-tunnel-attributes/p2p-primary-paths[name='*foo-path*']/lsp-instances[index='*local-index*']/state/notify-status	Sensor to export events for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs.
	
	ON_CHANGE support for LSP events is only activated when the reporting interval is set to 0 in the subscription request.
	
	This sensor is supported on indicated platforms up to and including Junos OS Release 17.3R1. See the following resource paths for LSP support in Junos OS Release 17.4R1 and higher:
	
	• /network-instances/network-instance[name='*instance-name*']/mpls/lsps/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/*p2p-primary-path*
	• /network-instances/network-instance[name='*instance-name*']/mpls/signaling-protocols/rsvp-te/sessions/session/state/notify-status
	
	Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1.
	
	The following events are exported under this resource path:
	
	• INITIATED
	• CONCLUDED_UP
	• CONCLUDED_TORN_DOWN
	• PROTECTION_AVAILABLE
	• PROTECTION_UNAVAILABLE
	• AUTOBW_SUCCESS
	• AUTOBW_FAIL
	• TUNNEL_LOCAL_REPAIRED
	• PATHERR_RECEIVED
	  • ADMISSION_CONTROL_FAILURE
	  • SESSION_PREEMPTED
	  • BAD_LOOSE_ROUTE
	  • BAD_STRICT_ROUTE
	  • LABEL_ALLOCATION_FAILURE
	  • ROUTING_LOOP_DETECTED
	  • REQUESTED_BANDWIDTH_UNAVAILABLE

Table 6: gRPC Sensors (continued)

resource path	Description
/mpls/lsps/constrained-path/tunnels/tunnel[name='*foo-name*',source='*foo-source*']/p2p-tunnel-attributes/p2p-primary-paths[name='*foo-path*']/state/notify-status	Sensor to export events for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs. ON_CHANGE support for LSP events is only activated when the reporting interval is set to 0 in the subscription request. This sensor is supported on indicated platforms up to and including Junos OS Release 17.3R1. See the following resource paths for LSP support in Junos OS Release 17.4R1 and higher: • /network-instances/network-instance[name='*instance-name*']/mpls/lsps/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/*p2p-primary-path* • /network-instances/network-instance[name='*instance-name*']/mpls/signaling-protocols/rsvp-te/sessions/session/state/notify-status Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1. The following events are exported under this resource path: • DESELECT_ACTIVE_PATH • CHANGE_ACTIVE_PATH • SELECT_ACTIVE_PATH • ORIGINATE_MBB • CSPF_NO_ROUTE • CSPF_SUCCESS • RESTART_RECOVERY_FAIL
/mpls/lsps/constrained-path/tunnels/tunnel[name='*foo-name*',source='*foo-source*']/p2p-tunnel-attributes/p2p-primary-paths[name='*foo-path*']/state/name	Sensor to export the path name for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs. This sensor is supported on indicated platforms up to and including Junos OS Release 17.3R1. See the following resource paths for LSP support in Junos OS Release 17.4R1 and higher: • /network-instances/network-instance[name='*instance-name*']/mpls/lsps/constrained-path/tunnels/tunnel/p2p-tunnel-attributes/p2p-primary-paths/ • /network-instances/network-instance[name='*instance-name*']/mpls/signaling-protocols/rsvp-te/sessions/session/state/notify-status Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1.

Table 6: gRPC Sensors (continued)

resource path	Description
/mpls/lsps/constrained-path/tunnels/tunnel[name='*foo-name*',source='*foo-source*']/p2p-tunnel-attributes/p2p-primary-paths[name='*foo-path*']/lsp-instances[index='*local-index*']/state/	Sensor to export LSP properties for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1. The following end paths are also supported for the resource path: • bandwidth • metric • max-average-bandwidth • explicit-route-objects • record-route-objects

/mpls/lsps/signaling-protocols/rsvp-te/sessions/session[local-index='*foo-index*']/state/notify-status	Sensor to export statistics for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs.
	
	ON_CHANGE support for LSP events is only activated when the reporting interval is set to 0 in the subscription request.
	
	Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1.
	
	The following events are exported under this resource path:
	
	• PATHERR_RECEIVED
	  • TTL_EXPIRED
	  • NON_RSVP_CAPABLE_ROUTER
	  • RESVTEAR_RECEIVED
	  • PATH_MTU_CHANGE

Table 6: gRPC Sensors (continued)

resource path	Description
/network-instances/network-instance/mpls/signaling-protocols/rsvp-te/sessions/session/state/notify-status	Supported on Junos OS Release through 17.4R1 and higher. Sensor to export events for ingress point-to-point LSPs, point-to-multipoint LSPs, bypass LSPs, and dynamically created LSPs. ON_CHANGE support for LSP events is only activated when the reporting interval is set to 0 in the subscription request. Supported on PTX Series routers, MX Series switches, and QFX10002, QFX10008, and QFX10016 switches starting with Junos OS Release 17.2R1. The following events are exported under this resource path: • **DETOUR-UP** • **DETOUR-DOWN** • **PATHERR-RECV** • **PATHERR-RECV/ADMISSION_CONTROL_FAILURE** • **PATHERR-RECV/SESSION_PREEMPTED** • **PATHERR-RECV/BAD_LOOSE_ROUTE** • **PATHERR-RECV/BAD_STRICT_ROUTE** • **PATHERR-RECV/LABEL_ALLOCATION_FAILURE** • **PATHERR-RECV/NON_RSVP_CAPABLE_ROUTER** • **PATHERR-RECV/TTL_EXPIRED** • **PATHERR-RECV/ROUTING_LOOP_DETECTED** • **PATHERR-RECV/REQUESTED_BANDWIDTH_UNAVAILABLE** • **PATHMTU-CHANGE**
/mpls/signaling-protocols/segment-routing/	Sensor for traffic statistics for both ingress IP traffic and transit MPLS traffic.. Supported on MX Series and PTX Series routers starting with Junos OS Release 18.3R1. The following end points are also supported and specify BGP Segment Routing traffic Engineering (SR-TE) transit statistics: • **/sr-te-bsid-policies/sr-te-bsid-policy[binding-sid='80001', to-address='*foo-to*' color='*foo-color*']/state/counters[name='oc-*xxx*']/packets** • **/sr-te-bsid-policies/sr-te-bsid-policy[binding-sid='80001', to-address='*foo-to*' color='*foo-color*']/state/counters[name='oc-*xxx*']/bytes** The following end points are also supported and specify BGP Segment Routing traffic Engineering (SR-TE) ingress statistics: • **/sr-te-ip-policies/sr-te-ip-policy[to-address='*foo-to*' color='*foo-color*']/state/counters[name='oc-*xxx*']/packets** • **/sr-te-ip-policies/sr-te-ip-policy[to-address='*foo-to*' color='*foo-color*']/state/counters[name='oc-*xxx*']/bytes** In addition to configuring the sensor, you must enable statistics collection using the statistics statement at the [**[edit protocols source-packet-routing telemetry statistics**] hierarchy level.

Table 6: gRPC Sensors (continued)

resource path	Description
/arp-information/	Sensor for Address Resolution Protocol (ARP) statistics for IPv4 routes. Supported on QFX10000 and QFX5200 switches switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers, EX9200 switches, and MX150 routers starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. Support on PTX10003 routers starting in Junos OS Evolved Release 19.1R1. You can also add the following to the end path for **/arp-information/** • **ipv4/neighbors/neighbor/state/ip** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **ipv4/neighbors/neighbor/state/link-layer-address** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **ipv4/neighbors/neighbor/state/origin** • **ipv4/neighbors/neighbor/state/host-name** • **ipv4/neighbors/neighbor/state/table-id** • **ipv4/neighbors/neighbor/state/neighbor-state** • **ipv4/neighbors/neighbor/state/expiry** • **ipv4/neighbors/neighbor/state/ispublish** • **ipv4/neighbors/neighbor/state/interface-name** • **ipv4/neighbors/neighbor/state/logical-router-id**

Table 6: gRPC Sensors (continued)

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

Table 6: gRPC Sensors (continued)

resource path	Description
	Sensor for Routing Engine internal interfaces. NOTE: On MX Series routers, you can specify the following interfaces: fxp0, em0, and em1 On PTX Series routers, you can specify the following interfaces: em0, ixlv0, ixlv1 On PTX Series routers with dual Routing Engines, you can specify the following interfaces: em0, ixgbe0, ixgbe1 On PTX10003 routers with Junos OS Evolved Release 19.1R1, you can specify the following interfaces: re0:mgmt-0. No internal interfaces are supported. Support on PTX1000 routers starting with Junos OS Release 17.3R1. Support on PTX10003 routers starting in Junos OS Evolved Release 19.1R1. The following end paths are also supported: • **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**

Table 6: gRPC Sensors (continued)

resource path	Description
	• /interfaces/interface/state/counters/out-sec-pkts • /interfaces/interface/state/counters/out-sec-octets • /interfaces/interface/state/counters/out-pause-pkts • /interfaces/interface/state/counters/in-drops • /interfaces/interface/state/counters/in-frame-errors • /interfaces/interface/state/counters/in-runts • /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 6: gRPC Sensors (continued)

resource path	Description
/nd6-information/	Sensor for Network Discovery Protocol (NDP) table state information for IPv6 routes. Supported on QFX10000 and QFX5200 switches starting with Junos OS Release 17.2R1. Supported on PTX1000 routers, EX9200 switches, and MX150 routers starting with Junos OS Release 17.3R1. Supported on QFX5100 switches starting with Junos OS Release 18.1R1. Supported on QFX5120-48Y and EX4650 switches starting with Junos OS Release 18.3R1. Supported on EX4600 switches starting with Junos OS Release 18.4R1. Supported on PTX10003 routers starting with Junos OS Evolved Release 19.1R1. You can also add the following to the end path for **nd6-information/** • **ipv6/neighbors/neighbor/state/ip** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **ipv6/neighbors/neighbor/state/link-layer-address** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **ipv6/neighbors/neighbor/state/origin** • **ipv6/neighbors/neighbor/sttae/is-router** ON_CHANGE streaming supported starting with Junos OS Evolved Release 19.1R1 • **ipv6/neighbors/neighbor/state/neighbor-state** • **ipv6/neighbors/neighbor/state/table-id** • **ipv6/neighbors/neighbor/state/is-secure** • **ipv6/neighbors/neighbor/state/is-publish** • **ipv6/neighbors/neighbor/state/expiry** • **ipv6/neighbors/neighbor/state/interface-name** • **ipv6/neighbors/neighbor/state/logical-router-id**
/ipv6-ra/	Sensor for NDP router-advertisement statistics.
/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. This sensor is supported starting on MX Series and PTX Series routers starting with Junos OS Release 17.4R1. Starting in Junos OS Evolved Release 19.1R1, PTX10003 routers are supported.

Table 6: 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 6: gRPC Sensors (continued)

resource path	Description
	Sensor for IS-IS routing protocol statistics. Statistics are exported separately for each routing instance. To specify a routing-instance name: **/network-instances/network-instance[name_'*instance-name*']/protocols/protocol/isis/levels/level/** **/network-instances/network-instance[name_'*instance-name*']/protocols/protocol/isis/interfaces/interface/levels/level/** NOTE: This sensor is supported on MX Series and PTX Series routers starting with Junos OS Release 17.4R1. The following paths are also supported: • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/received** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/processed** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/dropped** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/sent** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/lsp/retransmit** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/received** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/processed** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/dropped** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/sent** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/iih/retransmit** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/received** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/processed** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/dropped** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/sent** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/psnp/retransmit** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/cnsp/received** • **/network-instances/network-instance/protocols/protocol/isis/interfaces/interface/levels/level/packet-counters/cnsp/processed** • **/network-instances/network-instance/protocols/protocol/**

Table 6: 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 6: gRPC Sensors (continued)

resource path	Description
	• /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 6: gRPC Sensors (continued)

resource path	Description
	• /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 6: gRPC Sensors (continued)

resource path	Description
	• /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 6: 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 6: gRPC Sensors (continued)

resource path	Description
	Sensors for aggregate segment routing traffic with IS-IS. This sensor is supported on MX Series and PTX5000 routers starting with Junos OS Release 17.4R1. Statistics are exported separately for each routing instance. The first path exports inbound traffic. The second path exports outbound traffic. The third path exports inbound segment routing traffic for each segment identifier. 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. All MX and PTX5000 routers with FPC3 onwards support enhanced mode. If enhanced mode is not enabled, 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. 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. The following OpenConfig paths are supported: • **/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/**

Table 6: gRPC Sensors (continued)

resource path	Description
	sid-counters/sid-counter/forwarding-classes/forwarding-class/state/in-octets • **/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**
/junos/services/segment-routing/sid/usage/	Sensors for aggregate segment routing traffic with IS-IS. This sensor is supported on PTX3000 routers and PTX5000 routers with FPC2 starting with Junos OS Release 19.1R1. Statistics are exported separately for each routing instance. The first path exports inbound traffic. The second path exports outbound traffic. The third path exports inbound segment routing traffic for each segment identifier. 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.

Table 7: 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. You can also add the following to the end path for **/junos/system/subscriber-management/aaa/accounting-statistics/**: • **acct-req-received** • **acct-req-timeout** • **acct-resp-failure** • **acct-resp-success** • **acct-req-start** • **acct-req-interim** • **acct-req-stop** • **acct-resp-total** • **acct-resp-start** • **acct-resp-interim** • **acct-resp-stop** • **acct-resp-total**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/aaa/address-assignment-statistics/logical-system-routing-instances/logical-system-routing-instance/pools/pool	For Authentication, Authorization, and Accounting, this sensor tracks address pool utilization. The resource path can be refined to select a logical system routing instance by using a logical system routing instance filter: **/aaa/address-assignment-statistics/logical-system-routing-instances/logical-system-routing-instance [lsri-name=' *lsName:riName*']/pools/pool[pool-name=' *poolName*']** The resource path can be refined to select a specific pool by using a pool filter: **/junos/system/subscriber-management/aaa/address-assignment-statistics/logical-system-routing-instances/logical-system-routing-instance/pools/pool[pool-name=' *poolName*']** 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: **/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*']** The following end paths are also supported for the resource path: • **pool-name** • **out-of-memory** • **out-of-address** • **address-total** • **address-in-use** • **address-usage-percent** • **linked-pool-name**-The next address pool in the chain of linked pools. If there is no next address pool, the value is empty. • **linked-pool-head**-The first address pool in a chain of linked pools. If this pool is not part of a linked address pool, the value is empty.
/junos/system/subscriber-management/access-network/ancp/adapter	Sensors that track statistics associated with Access Node Control Protocol (ANCP) adapter. **mapped-dynamic-subscriber-count**—Number of ANCP subscribers mapped to dynamic interfaces by ANCP adapter.

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/access-network/ancp/protocol	Sensors that track statistics associated with ANCP protocol. **establishing-neighbor-count**—Number of neighbors in the process of establishing adjacency. **established-neighbor-count**—Number of neighbors in the process of establishing adjacency **total-neighbor-count**—Total number of neighbors in all states. **mapped-static-subscriber-count**—Number of ANCP subscribers mapped to static interfaces by ANCP protocol. **port-up-count**—Total number of port ups received. **port-down-count** —Total number of port downs received.
/junos/system/subscriber-management/aaa/authentication-statistics/	Sensors that track authentication, authorization, and accounting (AAA) authentication, pre-authentication, and re-authentication statistics. The following end paths are also supported for the resource path: • **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**
/junos/system/subscriber-management/aaa/dynamic-request-statistics/	Sensor tracks dynamic request statistics from AAA server-initiated requests, including Change of Authorization (CoA) and RADIUS-initiated Disconnect (RID). The following end paths are also supported for the resource path: • **dynamic-req-received** • **dynamic-req-success** • **dynamic-req-error** • **dynamic-req-silently-drop**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/aaa/radius-servers/radius-server/response-time/	Sensor for RADIUS server response time statistics for a specific server. 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**. 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. The resource path can be refined to select a specific RADIUS server by adding a server address filter to the resource path: **/junos/system/subscriber-management//aaa/radius-servers/radius-server[server-address='*radiusIpv4Address*']/response-time/** The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for RADIUS server statistics for a specific server. The resource path can be refined to select a specific RADIUS server by adding a server address filter to the resource path: **/junos/system/subscriber-management//aaa/radius-servers/ radius-server[server-address='*radiusIpv4Address*']/statistics/** The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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/	Sensor for DHVPv4 relay binding state statistics. The resource path can be refined to select a specific routing instance by adding a routing instance name: /junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name=' *routing-instance-name*]/relay/bindings/ The following end paths are also supported for the resource path: • 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/relay/servers/server/response-time	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. 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. 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. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name='** *routing-instance-name*]**/relay/servers/server/response-time** The resource path can be refined to select a specific DHCP server by adding a server filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/relay/servers/server[server-ip='** *server-ip*']**/response-time** 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: **/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** The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/server/bindings/	Sensor for DHVPv4 server binding state statistics. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[name='** *routing-instance-name* **]/server/bindings/** The following end paths are also supported for the resource path: • **binding-state-v4server-binding** • **binding-state-v4server-init** • **binding-state-v4server-bound** • **binding-state-v4server-selecting** • **binding-state-v4server-requesting** • **binding-state-v4server-renew** • **binding-state-v4server-release** • **binding-state-server-restoring**

Table 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for DHCPv4 telemetry for server statistics for a specific routing-instance. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='*routing-instance-name*']/server/statistics/** 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** 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 *routing-instance-name* is **default**. The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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/	Sensor for DHCPv4 telemetry. The following end paths are also supported for the resource path: • 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 • era-inflight-count • era-reported-failures • era-reported-successes

Table 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for DHVPv4 server statistics The resource path can be refined to select a specific routing instance by adding a routing instance name: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='*routing-instance-name*']/server/statistics/** 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** In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-ancpinstance[ri-name='*routing-instance-name*'] /server/statistics/** the only value supported for *routing-instance-name* is **default**. The following end paths are also supported for the resource path: • **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**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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

Table 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for DHVPv4 relay binding state statistics. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='*routing-instance-name*']/relay/statistics/** For example, the following resource path defines relay statistics for the **default:n000015k routing instance**: **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='n000015k]/relay/statistics** 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*']/relay/statistics/** the only value supported for the value *routing-instance-name* is **default**. The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/dhcp/v6/	Sensor for DHCPv6 statistics. The following end paths are also supported for the resource path: • **era-inflight-count** • **era-reported-failures** • **era-reported-successes**
/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/relay/bindings/	Sensor for DHVPv6 relay binding state statistics. The resource path can be refined to select a specific routing instance by adding a routing instance name: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='** *routing-instance-name*]**/relay/bindings/** The following end paths are also supported for the resource path: • **binding-state-v6relay-binding** • **binding-state-v6relay-init** • **binding-state-v6relay-bound** • **binding-state-v6relay-selecting** • **binding-state-v6relay-requesting** • **binding-state-v6relay-renew** • **binding-state-v6relay-release** • **binding-state-relay-restoring**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/relay/servers/server/response-time	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. 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. 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. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='** *routing-instance-name* **]/relay/servers/server/response-time** The resource path can be refined to select a specific DHCP server by adding a server address filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/relay/servers/server[server-ip='** *server-ip* **']/response-time** 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: **/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** The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance/server/bindings/	Sensor for DHVPv6 binding state statistics. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[name='** *routing-instance-name* **]/server/bindings/** The following end paths are also supported for the resource path: • **binding-state-v6server-binding** • **binding-state-v6server-init** • **binding-state-v6server-bound** • **binding-state-v6server-selecting** • **binding-state-v6server-requesting** • **binding-state-v6server-renew** • **binding-state-v6server-release** • **binding-state-server-restoring**

Table 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for DHCPv6 server statistics. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='*routing-instance-name*']/server/statistics/** For example, the following resource path defines server statistics for the **default:n000015k routing instance**: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='n000015k]/server/statistics** In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-protocols/dhcp/v6/ routing-instances/routing-instance[ri-name='*routing-instance-name*']/server/statistics** the only value supported for *routing-instance-name* is **default**. The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor for DHVPv6 relay statistics. The resource path can be refined to select a specific routing instance by adding a routing instance filter to the resource path: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='*routing-instance-name*']/relay/statistics/** For example, the following resource path defines relay statistics for the **default:n000015k routing instance**: **/junos/system/subscriber-management/client-protocols/dhcp/v6/routing-instances/routing-instance[ri-name='n000015k]/relay/statistics** In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-protocols/dhcp/v6/ routing-instances/routing-instance[ri-name='*routing-instance-name*']/relay/statistics** the only value supported for *routing-instance-name* is **default**. The following end paths are also supported for the resource path: • **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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/l2tp/summary/	Sensor for L2TP telemetry information. The following end paths are also supported for the resource path: • **l2tp-stats-total-tunnels** • **l2tp-stats-total-sessions** • **l2tp-stats-control-rx-packets** • **l2tp-stats-control-rx-bytes** • **l2tp-stats-control-tx-packets** • **l2tp-stats-control-tx-bytes** • **l2tp-era-type-icrq-inflight-count** • **l2tp-era-type-icrq-reported-successes** • **l2tp-era-type-icrq-reported-failures** • **l2tp-era-type-sccrq-inflight-count** • **l2tp-era-type-sccrq-reported-successes** • **l2tp-era-type-sccrq-reported-failures**
/junos/system/subscriber-management/client-protocols/ppp/statistics/	Sensors for PPP telemetry information. The following end paths are also supported for the resource path: • **ppp-stats-total-subscriber-sessions** • **ppp-stats-sessions-disable-phase** • **ppp-stats-sessions-establish-phase** • **ppp-stats-sessions-network-phase** • **ppp-stats-sessions-authenticate-phase**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/client-protocols/pppoe/statistics/	Sensors for PPPoE counts. The following end paths are also supported for the resource path: • **padi-packets-sent** • **padi-packets-received** • **pado-packets-sent** • **pado-packets-received** • **padr-packets-sent** • **padr-packets-received** • **pads-packets-sent** • **pads-packets-received** • **padt-packets-sent** • **padt-packets-received** • **service-error-sent** • **service-error-received** • **ac-error-sent** • **ac-error-received** • **generic-error-sent** • **generic-error-received** • **malformed-packets-received** • **unknown-packets-received** • **era-inflight-count** • **era-reported-successes** • **era-reported-failures**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber_management/dynamic-interfaces/interface-sets/meta-data/interface-set[container-id='*container-id-value*']/	Sensor for subscriber interface-set information. This sensor is supported on MX Series routers starting with Junos OS Release 18.4R1. ON-CHANGE streaming is supported. The following end paths are supported: • **cos-egress-tcp-name**-The egress traffic control profile associated with this **interface-set**. • **cos-egress-tcp-remainder-name**-The egress remainder traffic control profile associated with this **interface-set**. • **interface-set-name**-The name of the **interface-set** as supplied by AAA or as constructed by the topology relationship (ACI string or interface stacking). • **interface-set-type**-The type of interface-set (determines structure of interface-set-name). • **device-name**-The name of the underlying device or port (e.g. ge-1/0/0 or ae1). This leaf is empty if the **interface-set-type** is not a physical interface-set type. • **stag**-The outer VLAN tag. The value is 0 if **interface-set-type** is not a VLAN type. • **ctag**-The inner VLAN tag. The value is 0 if **interface-set-type** is not a VLAN type.
/junos/system/subscriber_management/dynamic-interfaces/interface-sets/meta-data/interface[sid-id='*sid-value*']/	Sensor for subscriber interface information. ON-CHANGE streaming is supported. The following end paths are supported: • **interface-index**-The system assigned interface index for the interface. • **session-type**-The type of client session (e.g VLAN, DHCP, PPPoE). • **user-name**-The login name for this interface and session. • **profile-name**-The name of the client profile used to create the interface. • **underlying-interface-name**-The name of the associated underlying interface. • **cvlan-tag**-The innermost VLAN tag value associated with the interface. • **svlan-tag**-The outermost VLAN tag value associated with the interface.

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber_management/dynamic-interfaces/interface-sets/meta-data/interface[sid-id='*sid-value*']/	Sensor for actual accounting statistics for dynamic subscriber interfaces. ON-CHANGE streaming is supported. The following end paths are supported: • **ip-in-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **ip-out-packets**-The number of actual transit IPv4 & IPv6 packets sent to the interface. • **ip-in-bytes**-The number of actual transit IPv4 & IPv6 bytes received by the interface. • **ip-out-bytes**-The number of actual transit IPv4 & IPv6 bytes received by the interface. • **ipv6-in-packets**-The number of actual transit IPv6 packets received by the interface. • **ipv6-out-packets**-The number of actual transit IPv6 packets sent to the interface. • **ipv6-in-bytes**-The number of actual transit IPv6 bytes received by the interface. • **ipv6-out-bytes**-The number of actual transit IPv6 bytes sent to the interface.
/junos/system/subscriber_management/dynamic-interfaces/interfaces/queue-statistics/interface[sid-id='*sid-value*']/fpcs/fpc[slot='*slot-value*']/queues/queue/[queue-no='*queue-no-value*']/	Sensor for queue statistics for dynamic interfaces. ON-CHANGE streaming is supported. The following end paths are supported: • **transmitted-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **transmitted-bytes**-Total bytes enqueued for this queue. • **dropped-packets**-Total packets dropped (because of RED, rate-limited, tail-drop, etc.) for the queue. • **dropped-bytes**-Total bytes dropped (because of RED, rate-limited, tail-drop, etc.) for the queue.
/junos/system/subscriber_management/dynamic-interfaces/interface-sets/queue-statistics/interface-set[container-id='*container-id-value*']/fpcs/fpc[slot='*slot-value*']/queues/queue/[queue-no='*queue-no-value*']/	Sensor for queue statistics for dynamic interface-sets. ON-CHANGE streaming is supported. The following end paths are supported: • **transmitted-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **transmitted-bytes**-Total bytes enqueued for this queue. • **dropped-packets**-Total packets dropped (because of RED, rate-limited, tail-drop, etc.) for the queue. • **dropped-bytes**-Total bytes dropped (because of RED, rate-limited, tail-drop, etc.) for the queue.

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/resource-monitor/chassis	Sensor for chassis resource statistics. The crossing of chassis thresholds maintained by the resource monitor can be incremented. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. Unless limits are configured using **configured-subscriber-limit**, configured and current limit counts will not be visible. The following end paths are supported for chassis threshold crossing statistics: • **subscriber-limit-exceeded** • **subscriber-limit-nominal** • **configured-subscriber-limit** • **current-subscriber-count**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/	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). The resource path can be refined to select a specific slot by adding a slot number filter to the resource path: **/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc[slot='** *slot number***']/statistics/** Using the slot number filter, the crossing of FPC thresholds maintained by the resource monitor can be incremented. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. Unless limits are configured using **configured-subscriber-limit**, configured and current limit counts will not be visible. The following end paths are supported for FPC threshold crossing statistics: • **mem-heap-exceeded** • **mem-heap-nominal** • **subscriber-limit-exceeded** • **subscriber-limit-nominal** • **configured-subscriber-limit** • **current-subscriber-count** The following end paths are also supported for the resource path: • **heap-memory-used** • **client-session-denied-count** • **service-session-denied-count** • **rtt-throttled-sub-count-client** • **rtt-throttled-sub-count-client**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/pfes/pfe	Sensor for FPC resource statistics at the Packet Forwarding Engine level. Periodically tracks line card statistics and Packet Forwarding Engine statistics. The resource path can be refined to select a specific Packet Forwarding Engine by adding a Packet forwarding Engine filter to the resource path: **/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc/statistics/pfes/pfe[pfe-no='** ***pfe number*****']/** The resource path can be refined to select a specific Packet Forwarding Engine by adding a slot number filter to the resource path: **/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc[slot='** ***slot number*****']/statistics/pfes/pfe[pfe-no='** ***pfe number*****']/** Using the slot number filter, the crossing of packet forwarding engine thresholds maintained by the resource monitor can be incremented. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. The following end paths are supported for packet forwarding threshold crossing statistics: • **mem-ifl-exceeded** • **mem-ifl-nominal** • **mem-expansion-exceeded** • **mem-expansion-nominal** The following end paths are also supported for the resource path: • **pfe-no** • **filter-memory-used** • **ifl-memory-used** • **expansion-memory-used** • **nh-memory**
/junos/system/subscriber-management/infra/resource-monitor/rsmon-infra/fpcs/fpc[slot=' ***slot number*****']/**	Sensor for FPC resource statistics. Using the slot number filter, the crossing of FPC thresholds maintained by the resource monitor can be incremented. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. The following end paths are supported for FPC threshold crossing statistics: • **delay-round-trip-exceeded** • **delay-round-trip-nominal**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc [slot='*slot number*']/statistics/pfes/pfe[pfe-no='*pfe number*']/sched-blocks/sched-block[sblock-no='*schedBlockNumber*']/	Sensor for counts of CoS utilization threshold crossing events above (exceeded) and below (nominal). For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. The following end paths are supported for CoS utilization threshold crossing statistics: • **cos-utilization-exceeded** • **cos-utilization-nominal** The following end paths are supported for statistical data: • **queues-max** • **queues-allocated**
/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc [slot='*slot number*']/pics/pic[pic-no='*pic number*']/	Sensor for PIC threshold crossing. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. Unless limits are configured using **configured-subscriber-limit**, configured and current limit counts will not be visible. The following end paths are supported for PIC threshold crossing statistics: • **subscriber-limit-exceeded** • **subscriber-limit-nominal** • **configured-subscriber-limit** • **current-subscriber-count**
/junos/system/subscriber-management/infra/resource-monitor/fpcs/fpc [slot='*slot number*']/pics/pic[pic-no='*pic number*']/ports/port[port-no='*port number*']/	Sensor for port threshold crossing. For each threshold, a count is maintained of rising and falling threshold crossings. As the consumed resource exceeds the threshold, the threshold exceeded count is incremented. As the consumed resource drops below the threshold, the threshold nominal count is incremented. Unless limits are configured using **configured-subscriber-limit**, configured and current limit counts will not be visible. The following end paths are supported for port utilization threshold crossing statistics: • **subscriber-limit-exceeded** • **subscriber-limit-nominal** • **configured-subscriber-limit** • **current-subscriber-count**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/network/dhcp/	Sensor for network stack DHCP. Periodically tracks packets processed by the BBE network stack to and from the DHCP application. The following end paths are also supported for the resource path: • **rx-packet-cnt** • **era-drops** • **rx-no-connection** • **rx-malformed-cnt** • **rx-no-if-cnt** • **rx-ifl-invalid** • **rx-send-failed** • **tx-packet-cnt** • **packets-transmitted** • **tx-malformed-cnt** • **tx-null-pkt** • **tx-no-if-cnt** • **tx-no-iff-cnt** • **tx-no-rtt-cnt** • **tx-arp-failed** • **tx_arp_failed** • **tx-if-invalid** • **tx-send-failed** • **rx-while-not-connected**
/junos/system/subscriber-management/infra/network/dvlan/	Sensor for network stack dynamic VLAN. Periodically maintains a count of the number of packets received that triggered dynamic VLAN interface creations. The following end paths are also supported for the resource path: • **rx-packet-cnt**

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/network/io/	Sensor for network stack IO. Periodcally provides basic network stack input and output and tracks network stack packet statistics. The following end paths are also supported for the resource path: • **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/	Sensor for network stack dynamic VLAN. Periodically maintains a count of the number of packets received that triggered dynamic VLAN interface creations. The following end paths are also supported for the resource path: • **rx-packet-cnt**

Table 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor network stack L2TP. Periodically tracks L2TP packets processed by the BBE network stack to and from the L2TP application. The following end paths are also supported for the resource path: • 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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 7: Broadband Edge gRPC Sensors (continued)

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

Table 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	Sensor network stack PPP. Periodically tracks PPP packets processed by the BBE network stack to and from the PPP application. The following end paths are also supported for the resource path: • 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
	• 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/	Sensor for network stack PPPoE statistics. PPPoE packets processed by the BBE network stack to and from the PPPoE application are tracked. The following end paths are also supported for the resource path: • 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/	Sensor for session database resources session counts by client type. The following end paths are also supported for the resource path: • 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 7: Broadband Edge gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/infra/sdb/statistics/state/	Sensor for session database resources tracking session counts by state. The following end paths are also supported for the resource path: • **init-state-count** • **configured-state-count** • **active-state-count** • **terminating-state-count** • **terminated-state-count** • **total-state-count**

Release History Table

Release	Description
19.1R1 EVO	Starting in Junos OS Evolved Release 19.1R1, OpenConfig (OC) and Junos Telemetry Interface (JTI) are supported. Both gRPC APIs and the customer-facing CLI remain the same as for the Junos OS. As was standard for Junos OS, Network Agent (NA) and OC packages are part of the Junos OS Evolved image.
19.1R1	Starting with Junos OS Evolved 19.1R1, Packet Forwarding Engine sensors on PTX10003 routers are also supported.
18.4R1	Starting with Junos OS Release 18.4R1, MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers are also supported.
18.3R1	Starting with Junos OS Release 18.3R1, ON_CHANGE streaming of LLDP telemetry sensor information is supported through gRPC for MX Series and PTX Series routers.
18.3R1	Starting with Junos OS Release 18.3R1, QFX5120-AY and EX4650 switches are also supported.
18.3R1	Starting with Junos OS Release 18.4R1, EX4600 switches are also supported.
18.2R1	Starting with Junos OS Release 18.2R1, PTX10002 routers are also supported.
18.1R1	Starting with Junos OS Release 18.1R1, QFX5210-64C switches and QFX5100 switches are also supported.
18.1R1	Starting with Junos OS Release 18.1R1, ON_CHANGE streaming of ARP, ND, and IP sensor information associated with interfaces is supported through gRPC for MX Series routers and PTX Series routers.
17.4R1	Starting with Junos OS Release 17.4R1, PTX10016 routers and virtual MX Series (vMX) routers are also supported.
17.3R1	Starting with Junos OS Release 17.3R1, QFX5110 switches, EX4600, EX4600-VC, and EX9200 switches and the Routing and Control Board (RCB) on 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 **/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance[ri-name='*routing-instance-name*'] /server/statistics/** the only value supported for *routing-instance-name* is **default**.
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-ancpinstance[ri-name='*routing-instance-name*'] /server/statistics/** the only value supported for *routing-instance-name* is **default**.
17.3R1	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*']/relay/statistics/** the only value

	supported for the value *routing-instance-name* is **default**.
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-protocols/dhcp/v6/ routing-instances/ routing-instance[ri-name='*routing-instance-name*']/server/statistics** the only value supported for *routing-instance-name* is **default**.
17.3R1	In Junos OS Release 17.3R1, broadband edge (BBE) gRPC sensor **/junos/system/subscriber-management/client-protocols/dhcp/v6/ routing-instances/ routing-instance[ri-name='*routing-instance-name*']/relay/statistics** the only value supported for *routing-instance-name* is **default**.
17.2R1	Starting with JunosOS Release 17.2R1, QFX10002, QFX10008, and QFX10016 switches, QFX5200 switches, and PTX1000 and PTX10008 routers are also supported.
16.1R3	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.

Related Documentation

- Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32

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*

Configure a Telemetry Sensor in Junos

Using Junos telemetry streaming, you can turn any available state information into a telemetry sensor by means of the XML Proxy functionality. The NETCONF XML management protocol and Junos XML API fully document all options for every supported Junos OS operational request. After you configure XML proxy sensors, you can access data over NETCONF "get" remote procedure calls (RPCs).

This task shows you how to stream the output of a Junos OS operational mode command.

BEST PRACTICE: We recommend that you not use YANG files that map to an 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.

This task requires the following:

- An MX Series, vMX Series, or PTX Series router operating Junos OS Release 17.3R2 or later.
- Installation of the required Network Agent package (network-agent-x86–32–17.4R1.16-C1.tgz or later).
- A telemetry data receiver, such as OpenNTI, to verify proper operation of your telemetry sensor.

In this task, you will stream the contents of the Junos OS command **show system users**.

show system users (vMX Series)

```
user@switch> show system users
USER     TTY      FROM                              LOGIN@  IDLE WHAT
user1   pts/0       172.31.12.36                    12:40PM     39 -cli (cli)

user2   pts/1       172,16.03.25                     3:01AM      - -cli (cli)
```

In addition to the expected list of currently logged-in users, the **show system users** output also provides the average system load as 1, 5 and 15 minutes. You can find the load averages by using the **show system users | display xml** command to view the XML tagging for the output fields. See **<load-average-1>**, **<load-average-5>**, and **<load-average-15>** in the XML tagging output below.

```
user@switch> show system users | display xml

<rpc-reply xmlns:junos="http://xml.juniper.net/junos/17.4R1/junos">
    <system-users-information xmlns="http://xml.juniper.net/junos/17.4R1/junos">

        <uptime-information>
            <date-time junos:seconds="1520170982">1:43PM</date-time>
            <up-time junos:seconds="86460">1 day, 40 mins</up-time>
            <active-user-count junos:format="2 users">2</active-user-count>
            <load-average-1>0.70</load-average-1>
            <load-average-5>0.58</load-average-5>
            <load-average-15>0.55</load-average-15>
            <user-table>
                <user-entry>
                    <user>root</user>
                    <tty>pts/0</tty>
                    <from>172.21.0.1</from>
                    <login-time junos:seconds="1520167202">12:40PM</login-time>
                    <idle-time junos:seconds="0">-</idle-time>
                    <command>cli</command>
                </user-entry>
                <user-entry>
                    <user>mwiget</user>
                    <tty>pts/1</tty>
                    <from>66.129.241.10</from>
                    <login-time junos:seconds="1520170862">1:41PM</login-time>
                    <idle-time junos:seconds="60">1</idle-time>
                    <command>cli</command>
                </user-entry>
            </user-table>
```

```
        </uptime-information>
    </system-users-information>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```

TIP: The uptime-information tag shown in the preceding output is a container that contains leafs, such as date-time, up-time, active-user-count. and load-average-1. Below is a sample YANG file for this container:

```
container uptime-information {
      dr:source "uptime-information"; // Exact name of the XML tag
      leaf date-time { // YANG model leaf
        type string; // Type of value
        dr:source date-time; // Exact name of the XML tag
      }
      leaf up-time { // YANG model leaf
        type string; // Type of value
        dr:source up-time; // Exact name of the XML tag
      }
      leaf active-user-count { // YANG model leaf
        type int32; // Type of value
        dr:source active-user-count; // Exact name of the XML tag
      }
      leaf load-average-1 { // YANG model leaf
        type string; // Type of value
        dr:source load-average-1; // Exact name of the XML tag
      }
      ...
```

TIP: The uptime-information tag also has another container named user-table that contains a list of user entries.

Below is a sample YANG file for this container:

```
container user-table { // "user-table" container which contains list of user-entry

        dr:source "user-table"; // Exact name of the XML tag
        list user-entry { // "user-entry" list which contains the users' details
 in form of leafs
          key "user"; // Key for the list "user-entry" which is a leaf in the
list "user-entry"
         dr:source "user-entry"; // Source of the list "user-entry" which is the
 exact name of the XML tag
          leaf user { // YANG model leaf
            dr:source user; // A leaf in the list "user-entry", exact name of the
 XML tag
            type string; // Type of value
          }
```

```
          leaf tty { // YANG model leaf
            dr:source tty; // A leaf in the list "user-entry", exact name of the
 XML tag
            type string; // Type of value
          }
          leaf from { // YANG model leaf
           dr:source from; // A leaf in the list "user-entry", exact name of the
 XML tag
            type string; // Type of value
          }
          leaf login-time { // YANG model leaf
            dr:source login-time; // A leaf in the list "user-entry", exact name
 of the XML tag
            type string; // Type of value
          }
          leaf idle-time { // YANG model leaf
            dr:source idle-time; // A leaf in the list "user-entry", exact name
of the XML tag
            type string; // Type of value
          }
          leaf command { // YANG model leaf
            dr:source command; // A leaf in the list "user-entry", exact name of
 the XML tag
            type string; // Type of value
          }
        }
      }
```


- Create a User-Defined YANG File on page 181
- Load the Yang File in Junos on page 184
- Collect Sensor Data on page 186
- Installing a User-Defined YANG File on page 188
- Troubleshoot Telemetry Sensors on page 189

Create a User-Defined YANG File

The YANG file defines the Junos CLI command to be executed, the resource path the sensors are placed under, and the key value pairs taken from the matching XML tags.

Custom YANG files for Junos OS conform to the YANG language syntax defined in RFC 6020 YANG 1.0 *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)* and RFC 7950 *The YANG 1.1 Data Modeling Language*. Certain directives need to be present in the file that configure XML proxy.

To use the **xmlproxyd** (daemon) process to translate telemetry data, create a ***render*.yang** file. In this file, the **dr:command-app** is set to **xmlproxyd**.

The XML proxy YANG filename and module name must start with **xmlproxyd_**:

- For the XML proxy YANG filename, add the extension **.yang**, for example, **xmlproxyd_sysusers.yang**
- For the module name, use the filename without the extension **.yang**, for example, **xmlproxyd_sysusers**

To simplify creating a YANG file, it's easiest to start by modifying a working example.

1. Provide a name for the module. The module name must start with **xmlproxyd_** and be the same name as the XML proxy YANG file name.

   For example, for an XML proxy YANG file called **sysusers.yang**, drop the **.yang** extension and name the module **xmlproxyd_sysusers**:

   module xmlproxyd_sysusers {

2. For the Junos Telemetry Interface, include the process (daemon) name **xmlproxyd**:

   dr:command-app "xmlproxyd";

3. Include the following RPC for the NETCONF get request:

   rpc juniper-netconf-get {

4. Specify the location of the output of the RPC, where *company-name* is the name you give to the location:

   dr:command-top-of-output "/company-name";

5. Include the following command to execute the RPC:

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

6. Specify the CLI command from which to retrieve data. The Junos OS CLI command that gets executed at the requested sample frequency is defined under **dr:cli-command** and executed by the **xmlproxyd** daemon.

To retrieve command output for the Junos OS command **show system users**:

dr:cli-command "show system users";

7. Escalate privileges, logon as “root”, connect to the internal management socket via Telnet, and specify help for an RPC:

   dr: command-help “default <get> rpc”;

   When this is included in the YANG file, output that is helpful for debugging is displayed in the **help drend** output on the internal management socket:

```
telnet /var/run/xmlproxyd_mgmt
Trying /var/run/xmlproxyd_mgmt...
Connected to /var/run/xmlproxyd_mgmt.
Escape character is '^]'.
220 XMLPROXYD release 18.2I20180412_0904_bijchand built by bijchand on
2018-04-12 14:48:48 UTC
help drend
```

200-juniper-netconf-get-0 system users <get> RPC

8. Specify the hierarchy and use the **dr:source** command to map to a container, a list, or a specific leaf. The absolute path under which the sensors will be reported is built from the output group **junos** plus **system-users-information**, concatenated by **/'**. The path **/junos/system-users-information/** is the path to query for information about this custom sensor.

   WARNING: You should not create a custom YANG model that conflicts or overlaps with predefined native paths (Juniper defined paths) and OpenConfig paths (resources). Doing so can result in undefined behavior.

   For example, do not create a model that defines new leafs at or augments nodes for resource paths such as **/junos/system/linecard/firewall**or **/interfaces**.

   A one-to-one mapping between container, leafs and the XML tag or value from the CLI command output is defined in the grouping referenced by **uses** within the output container. A *grouping* can be referred to multiple times in different container outputs. The container **system-users-information** below uses the grouping system-users-information. However, it is defined without the aforementioned one-to-one mapping for every container, list and leaf to an output XML tag from the CLI command XML output.

```
output {
  container junos {
    container system-users-information {
      dr:source "/system-users-information“;
      uses system-users-information-grouping;
```

```
    }
  }
}
```

9. The following YANG file shows how to include these commands to enable the **xmlproxyd** process to retrieve the full operational state and map it to the leafs in Juniper's own data model:

```
*/
```

```
/*
 * Example yang for generating OpenConfig equivalent of show system users
 */

module xmlproxyd_sysusers {
  yang-version 1;

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

  import drend {
    prefix dr;
  }

  grouping system-users-information-grouping {
    container uptime-information {
      dr:source "uptime-information";
      leaf date-time {
        type string;
        dr:source date-time;
      }
      leaf up-time {
        type string;
        dr:source up-time;
      }
      leaf active-user-count {
        type int32;
        dr:source active-user-count;
      }
      leaf load-average-1 {
        type string;
        dr:source load-average-1;
      }
      leaf load-average-5 {
        type string;
        dr:source load-average-5;
      }
      leaf load-average-15 {
        type string;
        dr:source load-average-15;
      }
      container user-table {
        dr:source "user-table";
        list user-entry {
          key "user";
          dr:source "user-entry";
          leaf user {
            dr:source user;
```

```
          type string;
        }
        leaf tty {
          dr:source tty;
          type string;
        }
        leaf from {
          dr:source from;
          type string;
        }
        leaf login-time {
          dr:source login-time;
          type string;
        }
        leaf idle-time {
          dr:source idle-time;
          type string;
        }
        leaf command {
          dr:source command;
          type string;
        }
      }
    }
  }
}

dr:command-app "xmlproxyd";
rpc juniper-netconf-get {
  dr:command-top-of-output "/company-name";
  dr:command-full-name "drend juniper-netconf-get";
  dr:cli-command "show system users";
  dr:command-help "default <get> rpc";
```

```
output {
      container company-name {
        container system-users-information {
          dr:source "/system-users-information";
          uses system-users-information-grouping;
        }
      }
    }
  }
}
```

Load the Yang File in Junos

After the YANG file is complete, upload the YANG file and verify that the module is created.

1. Upload the YANG file to the router.

2. Register the YANG file using the **request system yang add package** command.

   ```
   user@switch> request system yang add package sysusers proxy-xml module
   xmlproxyd_sysusers.yang
   ```

```
XML proxy YANG module validation for xmlproxyd_sysusers.yang : START
XML proxy YANG module validation for xmlproxyd_sysusers.yang : SUCCESS
JSON generation for xmlproxyd_sysusers.yang : START
JSON generation for xmlproxyd_sysusers.yang: SUCCESS
```

NOTE: Starting in Junos OS Release 18.3R1, adding, deleting, or updating YANG packages in configuration mode with the run command is not supported.

3. Verify that the module (sensor) is registered using the **show system yang package sysusers** command, where **sysusers** is the name of the package:

```
user@switch> show system yang package sysusers
Package ID              :sysusers
XML Proxy YANG Module(s) :xmlproxyd_sysusers.yang
```

4. Enable gRPC in the Junos OS configuration:

```
user@switch> set system services extension-service request-response grpc port 32767
```

Collect Sensor Data

Use your favorite collector to pull the newly created telemetry sensor data from the device. .The following instructions use the collector *jtimon*. For information about jtimon setup, see Junos Telemetry Interface client.

1. Create a simple configuration file, here named **vmx1.json**. Adjust the host IP address and the port, as needed. The path **/junos/system-users-information** is specified. The **freq** field is defined in MicroSoft, streaming a new set of key value pairs every 5 seconds. Optionally, you can add multiple paths.

```
$ cat vmx1.json
{
  "host": "172.16.122.182
  "port": 32767
  "cid": "my-client-id",
  "grpc" : {
    "ws" : 524289
  },
  "paths": {
     {
       "path": "/junos/system-users-information/",
       "freq": 5000
     },
     {
       "path": "/junos/additional-path/",  <-OPTIONAL
       "freq": 5000
     }
   }
}
```

2. Launch the collector, using either your own compiled file or an automatically built image from Docker Hub.
   The sample query output below shows the sensor report by path. Every key is sent in human-readable form as an absolute path. In case of lists, the absolute path contains an index in the form of XPATH which is ideal to group values from a (time series) database, such as InfluxDB. For example, the output below shows the path **/junos/system-users-information/uptime-information/user-table/user-entry[user='ab']/**.

   You can terminate the stream of sensor data using Ctrl-C.

   $ docker run -tu --rm -v $(PWD):/u mw/jtimon --config vmx1.json --print

```
gRPC headers from Junos:
  init-response: [response { subscription_id 1} path_list {path:
"junos/system-users-information/" sample-frequency: 5000 } ]
  content-type: [application/grpc]
  grpc-accept-encoding: [identity,deflate,gzip]
2018/03/04 17:13:19 system-id vmxdockerlight_vmx1_1
2018/03/04 17:13:19 component_id 65535
2018/03/04 17:13:19 sub_component_id: 0
2018/03/04 17:13:19 path:
sensor_1000:/junos/system-users-information/:/junos/system-users-information/
2018/03/04 17:13:19 sequence_number: 16689
2018/03/04 17:13:19 timestamp: 1520183589391
```

```
2018/03/04 17:13:19 sync_response: %!d(bool=false)
2018/03/04 17:13:19 key: __timestamp__
2018/03/04 17:13:19 uint_value: 1520183589391
2018/03/04 17:13:19 key: __junos_re_stream_creation_timestamp--
2018/03/04 17:13:19 uint value: 1520183589372
2018/03/04 17:13:19 key: __junos_re_payload-get_timestamp__
2018/03/04 17:13:19 uint_value: 1520183589390
2018/03/04 17:13:19 key:
/junos/system-users-information/uptime-information/date-time
2018/03/04 17:13:19 str-value: 5:13PM
2018/03/04 17:13:19 key:
/junos/system-users-inforamtion/uptime-information/up-time
2018/03/04 17:13:19 str-value: 1 day, 4:10
2018/03/04 17:13:19 key:
/junos/system-users-information/uptime-information/active-user-count
2018/03/04 17:13:19 int_value: 2
2018/03/04 17:13:19 key:
/junos/system-users-inforamtion/uptime-information/load-average-1
2018/03/04 17:13:19 str_value: 0.62
2018/03/04 17:13:19 key:
/junos/system-users-information/uptime-information/load-average-5
2018/03/04 17:13:19 str_value: 0.56
2018/03/04 17:13:19 key:
/junos/system-users-inforamtion/uptime-information/load-average-15
2018/03/04 17:13:19 str_value: 0.53
2018/03/04 17:13:19 key: __prefix__
2018/03/04 17:13:19 str_value:
/junos/system-users-information/uptime-information/user-table/user-entry[user='ab']/
2018/03/04 17:13:19 key: tty
2018/03/04 17:13:19 str_value: pts/1
2018/03/04 17:13:19 key: from
2018/03/04 17:13:19 str-value: 172,16.04.25
2018/03/04 17:13:19 key: login-time
2018/03/04 17:13:19 str_value: 5:12PM
2018/03/04 17:13:19 key: idle-time
2018/03/04 17:13:19 str-value: -
2018/03/04 17:13:19 key: command
2018/03/04 17:13:19 str_value: -cl
2018/03/04 17:13:19 system_id: vmxdockerlight_vmx1_1
2018/03/04 17:13:19 component_id: 65535
2018/03/04 17:13:19 sub_component_id: 0
2018/03/04 17:13:19 <output truncated>
```

The sample query shown below shows two sensor reports per path, then I terminated it with Ctrl-C. Every key is sent in human readable form as an absolute path and in case of lists, contains an index in form of XPATH, ideal to group values from a (time series) database like InfluxDB e.g. /junos/system-users-information/uptime-information/user-table/user-entry[user='ab']/

3. Verify that the module (sensor) is loaded using the **show system yang package sysusers** command, where **sysusers** is the name of the package:

```
user@switch> show system yang package sysusers
```

```
Package ID               :sysusers
XML Proxy YANG Module(s) :xmlproxyd_sysusers.yang
```

4. Enable gRPC in the Junos OS configuration:

```
user@switch> set system services extension-service request-response grpc port 32767
```

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 XML proxy YANG file and the file path to install it. This command creates a **.json** file in the **/opt/lib/render** directory.

```
user@switch> 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-name* in brackets: **[** *file-path-name 1 file-path-name 2* **]**

2. (Optional) Validate an module before adding it to the router using the **request system yang validate proxy-xml module** *module-name* command. .

```
user@switch> request system yang validate proxy-xml module module-name
```

The output **XML proxy YANG module validation for xmlproxyd_<module-name> : SUCCESS** indicates successful module validation.

Mismatch error sometimes occur. If the command returns the error below, you can eliminate the error by using Junos OS Release 17.3R2 or later:

```
user@switch> request system yang validate proxy-xml module
  xmlproxyd_sysusers.yang
  error: illegal identifier <identifier> , must not start with [xX][mM][lL]
```

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

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

4. Delete an existing XML proxy YANG file.

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

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

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

See Also

- Understanding YANG on Devices Running Junos OS on page 176
- Installing the Network Agent Package (Junos Telemetry Interface) on page 45
- Guidelines for gRPC Sensors (Junos Telemetry Interface) on page 51
- *Sending Requests to the NETCONF Server*

Troubleshoot Telemetry Sensors

Problem **Description:** Use the following methods to troubleshoot user-define telemetry sensors:

- Execute a tcpdump for the interface your gRPC requests came from (for this task, interface **fxp0** was used).

```
user@switch>monitor traffic interface fxp0 no-resolve matching "tcp port 32767"
```

- Enable traceoptions using the **set services analytics traceoptions flag xmlproxy** command. Check the **xmlproxyd** log file for confirmation of whether the CLI command's RPC was sent and if a response was received:

1. Issue the **show log xmlproxyd** command to show the xmlproxyd log. The value for the field **xmlproxy_execute_cli_command:** indicates if the RPC was sent or not. The value for the field **xmlproxy_build_context** indicates the command.

```
user@switch>show log xmlproxyd
Mar 4 18:52:46 vmxdockerlight_vmx1_1 clear-log[52495]: logfile cleared
Mar  4 18:52:51 xmlproxy_telemetry_start_streaming: sensor
/junos/system-users-information/
Mar  4 18:52:51 xmlproxy_build_context: command show system users merge-tag:
Mar  4 18:52:51 <command format="xml">show system users</command>
Mar  4 18:52:51 xmlproxy_execute_cli_command: Sent RPC..
Mar  4 18:52:51 <system-users-information
xmlns="http://xml.juniper.net/junos/17.4R1/junos"
xmlns:junos="http://xml.juniper.net/junos/*/junos">
<uptime-information>
<date-time junos:seconds="1520189571">
6:52PM
</date-time>
<up-time junos:seconds="107400">
1 day,  5:50
</up-time>
<active-user-count junos:format="1 users">
1
</active-user-count>
<load-average-1>
0.94
</load-average-1>
<load-average-5>
```

```
0.73
</load-average-5>
<load-average-15>
0.65
```

See Also

- Understanding YANG on Devices Running Junos OS on page 176
- Installing the Network Agent Package (Junos Telemetry Interface) on page 45
- *Configurable NETCONF Proxy for Junos Telemetry Interface*
- Guidelines for gRPC Sensors (Junos Telemetry Interface) on page 51
- *Sending Requests to the NETCONF Server*

Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Enable Export of Subscriber Statistics and Queue Statistics on page 192
- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface) on page 194

Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

You can use subscriber statistics and queue statistics for dynamic interfaces and interface-sets to support remote analytics and monitoring on Juniper devices that operate as a Broadband Network Gateway (BNG). Using these statistics, you can model and condition traffic flows in a subscriber access network.

- About Subscriber and Queue Statistics on page 190
- Enabling Export of Statistics on page 191

About Subscriber and Queue Statistics

Subscriber statistics include the per IP protocol family (IPv4 or IPv6) packet information (receive and transmitted packets and bytes) for a subscriber interface. They will only include subscriber data forwarded by the system. Filtered and dropped packets and control traffic are factored out and not delivered.

ON-CHANGE subscription support for interface meta-data sends asynchronous notifications when interfaces are created and deleted. After an initial baseline of delivering **create** notifications for all existing interfaces, only notifications for interfaces that are being created or deleted are sent to an external collector.

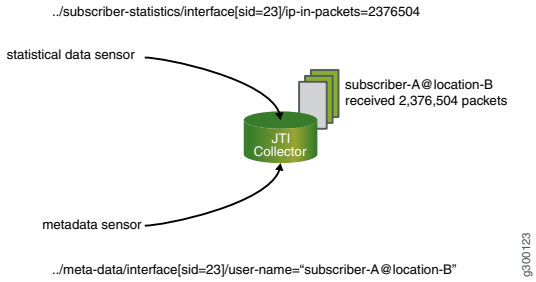
Use queue statistics to determine oversubscription levels, the mix of forwarding-class traffic, or traffic rates for a given CoS-enabled interface or interface-set.

Enabling Export of Statistics

To receive statistics, you enable both meta-data and statistical data for export on your Juniper device through the Junos CLI. Meta-data for the interface is provided because the interface key is a dynamic integer, a session identifier (SID), which conveys no context to an external server. The meta-data provides more tangible context (such as the user name, a profile name VLAN tags, etc.) to the SID. An external collector associates the statistical data to a persistent reference.

A subscription for both statistical data and meta-data can be made from the external collector (in Figure 2 on page 191, the JTI collector). In this way, the two streams are "merged" and a correlation is made between the statistical data and the meta-data. The dynamic SID is matched with the more permanent attributes such as user name and location.

Figure 2: JTI Collector "Merging" Sensor Data



See Also

- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- Enable Export of Subscriber Statistics and Queue Statistics on page 192
- gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface) on page 194
- telemetry on page 277

Enable Export of Subscriber Statistics and Queue Statistics

You can enable the telemetry export of subscriber statistics and queue statistics for dynamic interfaces and interface-sets. After you enable telemetry for these statistics, they are eligible for export to one or more collectors using a remote procedure call (gRPC) subscription.

Use these statistics to model and condition traffic flows in a subscriber access network and to provide subscriber statistics information (accurate accounting).

To enable the export of subscriber statistics and associated interface meta-data:

1. Enable export of interface meta-data and subscriber statistics:

```
[edit dynamic-profiles profile-name]
user@host# set telemetry subscriber-statistics
```

2. Enable the logical demultiplexing (demux) interface in a dynamic profile to export subscriber accurate statistics:

```
[edit dynamic-profiles interfaces demux0]
user@host# unit $junos-interface-unit actual-transit-statistics
```

To enable export of interface meta-data and queue statistics for dynamic interfaces:

1. Enable export of interface meta-data and interface queue statistics. Use the profile variable **$junos-interface-name**.

NOTE: the profile variables $junos-interface-name and $junos-interface-set-name are generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time. Using these derived variables is a convenient way to configure telemetry behavior for the interface or interface-set without the need to mimic the specific configuration in the interfaces stanza.

```
[edit dynamic-profiles profile-name]
user@host# set telemetry queue-statistics interface $junos-interface-name
```

2. To override the default internal queue-stats collection interval of 900 seconds or the default queue export filter (all queues, 0-7), add the **rate** and **queues** statements.

```
[edit dynamic-profiles profile-name telemetry queue-statistics interface
   $junos-interface-name]
user@host# set rate 300
user@host# set queues “0,1,2”
```

To enable export of interface-set meta-data and queue statistics for dynamic interface-sets:

1. Enable export of interface-set meta-data and interface-set queue statistics. Use the profile variable **$junos-interface-set-name**.

   NOTE: the profile variables **$junos-interface-name** and **$junos-interface-set-name** are generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time. Using these derived variables is a convenient way to configure telemetry behavior for the interface or interface-set without the need to mimic the specific configuration in the interfaces stanza.

   ```
   [edit dynamic-profiles profile-name]
   user@host# set telemetry queue-statistics interface-set $junos-interface-set-name
   ```

2. To override the default internal queue-stats collection interval of 900 seconds or the default queue export filter (all queues, 0-7), add the **rate** and **queues** statements.

   ```
   [edit dynamic-profiles profile-name telemetry queue-statistics interface-set
     $junos-interface-set-name]
   user@host# set rate 300
   user@host# set queues "0,1,2"
   ```

After telemetry export is enabled, meta-data and statistics can be streamed to external collectors subscribing to the available resource paths.

Use the resource paths from "gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface)" on page 194 for your gRPC subscription.

See Also

- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface) on page 194
- Configure a Telemetry Sensor in Junos on page 177

Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

You can use subscriber statistics and queue statistics for dynamic interfaces and interface-sets to support remote analytics and monitoring on MX Series routers that operate as a Broadband Network Gateway (BNG).

Before enabling export of subscriber statistics and queue statistics for dynamic interfaces and interface-sets, consider the following limitations:

- On MX Series routers supporting the Modular Port Concentrator 2 (MPC2), a slow internal refresh cycle for queue statistics can occur. This cycle can be lengthy at full line card scale. If the subscription frequency is higher than the internal refresh cycle, exported data may appear stale across reporting intervals.
- The unified in-service software upgrade (ISSU) feature enables you to upgrade your device between two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic. Dynamic interfaces and Interface-sets created prior to ISSU and prior to Junos OS Release 18.4R1 do not support telemetry for subscriber and queue statistics.
- The subscription frequency should be larger than the time to export telemetry. If the volume of data cannot be exported before the next reporting interval, the export continues to completion and the next reporting interval will immediately start. in such instances, continuous streaming results–behavior that may not be wanted.
- Multiple sensors from the dynamic-interfces sub-tree may be subscribed to simultaneously. As streaming of these sensors for the sub-tree is supported by a single Junos component, you should expect the time to export the sensor data for each subscription to extend.
- Juniper advises to enable export only for active queues. To do this, include the **queues** statement at the [**[edit dynamic-profiles *profile-name* telemetryqueue-statistics $junos-interface-name]** or [**[edit dynamic-profiles *profile-name* telemetry queue-statistics $junos-interface-set-name]** hierarchy level. Exporting data for active queues only reduces the amount of data to export for each reporting interval.

See Also

- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Enable Export of Subscriber Statistics and Queue Statistics on page 192

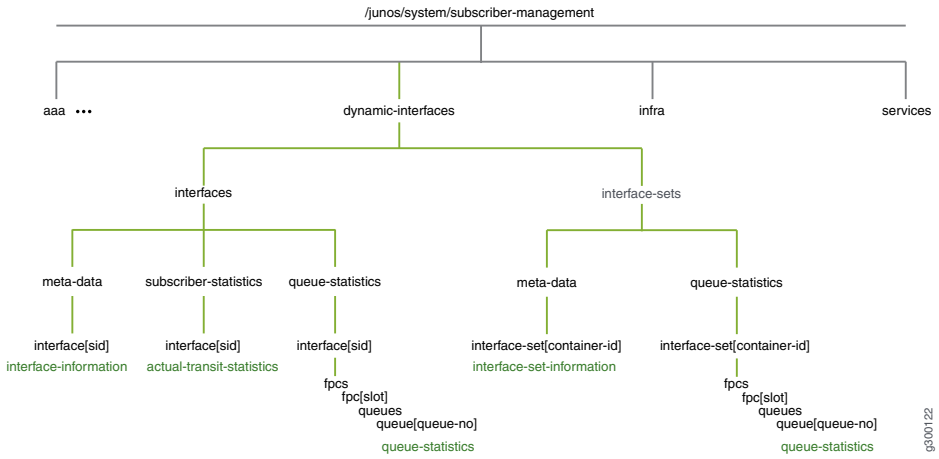
gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface)

Starting with Junos OS Release 18.4R1, MX Series routers are supported.

You can use subscriber statistics and queue statistics for dynamic interfaces and interface-sets to support remote analytics and monitoring on Juniper devices that operate as a Broadband Network Gateway (BNG). Using these statistics, you can model and condition traffic flows in a subscriber access network.

Figure 3 on page 195 shows the structure of the sensors or resource paths used for subscription to the external collector. The resource paths are a combination of both meta-data and statistical data.

Figure 3: Structure of Sensors



For statistics delivery through a gRPC subscription, include one or more resource paths from Table 6 on page 53 in the subscription. For statistics delivered through gRPC, you will also need to install some additional software an enable statistics to be exported on your Juniper device through the Junos CLI. For more information, see "Enable Export of Subscriber Statistics and Queue Statistics" on page 192. For more information about creating a subscription, see "Configure a Telemetry Sensor in Junos" on page 177.

Table 8: gRPC Sensors

resource path	Description
/junos/system/subscriber-management/ dynamic-interfaces/interface-sets/meta-data/ interface-set[container-id='*container-id-value*']/	Sensor for subscriber interface-set information. This sensor is supported on MX Series routers starting with Junos OS Release 18.4R1. ON-CHANGE streaming is supported. The following end paths are supported: • **cos-egress-tcp-name**-The egress traffic control profile associated with this **interface-set**. • **cos-egress-tcp-remainder-name**-The egress remainder traffic control profile associated with this **interface-set**. • **interface-set-name**-The name of the **interface-set** as supplied by AAA or as constructed by the topology relationship (ACI string or interface stacking). • **interface-set-type**-The type of interface-set (determines structure of interface-set-name). • **device-name**-The name of the underlying device or port (e.g. ge-1/0/0 or ae1). This leaf is empty if the **interface-set-type** is not a physical interface-set type. • **stag**-The outer VLAN tag. The value is 0 if **interface-set-type** is not a VLAN type. • **ctag**-The inner VLAN tag. The value is 0 if **interface-set-type** is not a VLAN type.

Table 8: gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/dynamic-interfaces/interfaces/meta-data/interface[sid='*sid-value*']/	Sensor for subscriber interface information. ON-CHANGE streaming is supported. The following end paths are supported: • **interface-index**-The system assigned interface index for the interface. • **session-type**-The type of client session (e.g VLAN, DHCP, PPPoE). • **user-name**-The login name for this interface and session. • **profile-name**-The name of the client profile used to create the interface. • **underlying-interface-name**-The name of the associated underlying interface. • **cvlan-tag**-The innermost VLAN tag value associated with the interface. • **svlan-tag**-The outermost VLAN tag value associated with the interface.
/junos/system/subscriber-management/dynamic-interfaces/interfaces/subscriber-statistics/interface[sid='*sid-value*']/	Sensor for actual accounting statistics for dynamic subscriber interfaces. The following end paths are supported: • **ip-in-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **ip-out-packets**-The number of actual transit IPv4 & IPv6 packets sent to the interface. • **ip-in-bytes**-The number of actual transit IPv4 & IPv6 bytes received by the interface. • **ip-out-bytes**-The number of actual transit IPv4 & IPv6 bytes received by the interface. • **ipv6-in-packets**-The number of actual transit IPv6 packets received by the interface. • **ipv6-out-packets**-The number of actual transit IPv6 packets sent to the interface. • **ipv6-in-bytes**-The number of actual transit IPv6 bytes received by the interface. • **ipv6-out-bytes**-The number of actual transit IPv6 bytes sent to the interface.
/junos/system/subscriber-management/dynamic-interfaces/interfaces/queue-statistics/interface[sid='*sid-value*']/fpcs/fpc[slot='*slot-value*']/queues/queue/[queue-no='*queue-no-value*']/	Sensor for queue statistics for dynamic interfaces. The following end paths are supported: • **transmitted-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **transmitted-bytes**-Total bytes enqueued for this queue. • **dropped-packets**-Total packets dropped (because of RED, rate-limited, tail-drop, etc.) for the queue. • **dropped-bytes**-Total bytes dropped (because of RED, rate-limited, tail-drop, etc.) for the queue.

Table 8: gRPC Sensors (continued)

resource path	Description
/junos/system/subscriber-management/dynamic-interfaces/interface-sets/queue-statistics/interface-set[container-id='*container-id-value*']/fpcs/fpc[slot='*slot-value*']/queues/queue/[queue-no='*queue-no-value*']/	Sensor for queue statistics for dynamic interface-sets. The following end paths are supported: • **transmitted-packets**-The number of actual transit IPv4 & IPv6 packets received by the interface. • **transmitted-bytes**-Total bytes enqueued for this queue. • **dropped-packets**-Total packets dropped (because of RED, rate-limited, tail-drop, etc.) for the queue. • **dropped-bytes**-Total bytes dropped (because of RED, rate-limited, tail-drop, etc.) for the queue.

See Also • Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32

Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

You can use subscriber statistics and queue statistics for dynamic interfaces and interface-sets to support remote analytics and monitoring on Juniper devices that operate as a Broadband Network Gateway (BNG). Using these statistics, you can model and condition traffic flows in a subscriber access network.

- About Subscriber and Queue Statistics on page 197
- Enabling Export of Statistics on page 197

About Subscriber and Queue Statistics

Subscriber statistics include the per IP protocol family (IPv4 or IPv6) packet information (receive and transmitted packets and bytes) for a subscriber interface. They will only include subscriber data forwarded by the system. Filtered and dropped packets and control traffic are factored out and not delivered.

ON-CHANGE subscription support for interface meta-data sends asynchronous notifications when interfaces are created and deleted. After an initial baseline of delivering **create** notifications for all existing interfaces, only notifications for interfaces that are being created or deleted are sent to an external collector.

Use queue statistics to determine oversubscription levels, the mix of forwarding-class traffic, or traffic rates for a given CoS-enabled interface or interface-set.

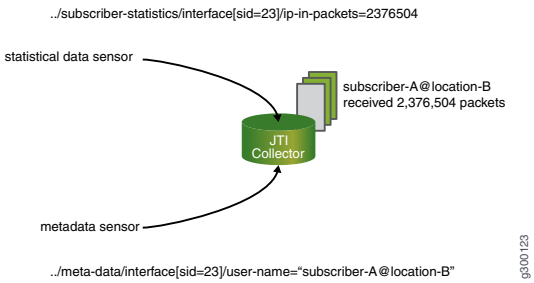
Enabling Export of Statistics

To receive statistics, you enable both meta-data and statistical data for export on your Juniper device through the Junos CLI. Meta-data for the interface is provided because the interface key is a dynamic integer, a session identifier (SID), which conveys no context to an external server. The meta-data provides more tangible context (such as the user

name, a profile name VLAN tags, etc.) to the SID. An external collector associates the statistical data to a persistent reference.

A subscription for both statistical data and meta-data can be made from the external collector (in Figure 2 on page 191, the JTI collector). In this way, the two streams are "merged" and a correlation is made between the statistical data and the meta-data. The dynamic SID is matched with the more permanent attributes such as user name and location.

Figure 4: JTI Collector "Merging" Sensor Data



Related Documentation

- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- Enable Export of Subscriber Statistics and Queue Statistics on page 192
- gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface) on page 194
- telemetry on page 277

Enable Export of Subscriber Statistics and Queue Statistics

You can enable the telemetry export of subscriber statistics and queue statistics for dynamic interfaces and interface-sets. After you enable telemetry for these statistics, they are eligible for export to one or more collectors using a remote procedure call (gRPC) subscription.

Use these statistics to model and condition traffic flows in a subscriber access network and to provide subscriber statistics information (accurate accounting).

To enable the export of subscriber statistics and associated interface meta-data:

1. Enable export of interface meta-data and subscriber statistics:

   ```
   [edit dynamic-profiles profile-name]
   user@host# set telemetry subscriber-statistics
   ```

2. Enable the logical demultiplexing (demux) interface in a dynamic profile to export subscriber accurate statistics:

```
[edit dynamic-profiles interfaces demux0]
user@host# unit $junos-interface-unit actual-transit-statistics
```

To enable export of interface meta-data and queue statistics for dynamic interfaces:

1. Enable export of interface meta-data and interface queue statistics. Use the profile variable **$junos-interface-name**.

   NOTE: the profile variables $junos-interface-name and $junos-interface-set-name are generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time. Using these derived variables is a convenient way to configure telemetry behavior for the interface or interface-set without the need to mimic the specific configuration in the interfaces stanza.

   ```
   [edit dynamic-profiles profile-name]
   user@host# set telemetry queue-statistics interface $junos-interface-name
   ```

2. To override the default internal queue-stats collection interval of 900 seconds or the default queue export filter (all queues, 0-7), add the **rate** and **queues** statements.

   ```
   [edit dynamic-profiles profile-name telemetry queue-statistics interface
     $junos-interface-name]
   user@host# set rate 300
   user@host# set queues “0,1,2”
   ```

To enable export of interface-set meta-data and queue statistics for dynamic interface-sets:

1. Enable export of interface-set meta-data and interface-set queue statistics. Use the profile variable **$junos-interface-set-name**.

   NOTE: the profile variables $junos-interface-name and $junos-interface-set-name are generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time. Using these derived variables is a convenient way to configure telemetry behavior for the interface or interface-set without the need to mimic the specific configuration in the interfaces stanza.

   ```
   [edit dynamic-profiles profile-name]
   user@host# set telemetry queue-statistics interface-set $junos-interface-set-name
   ```

2. To override the default internal queue-stats collection interval of 900 seconds or the default queue export filter (all queues, 0-7), add the **rate** and **queues** statements.

```
[edit dynamic-profiles profile-name telemetry queue-statistics interface-set
   $junos-interface-set-name]
user@host# set rate 300
user@host# set queues “0,1,2”
```

After telemetry export is enabled, meta-data and statistics can be streamed to external collectors subscribing to the available resource paths.

Use the resource paths from “gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface)” on page 194 for your gRPC subscription.

Related Documentation

- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- gRPC Sensors for Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets (Junos Telemetry Interface) on page 194
- Configure a Telemetry Sensor in Junos on page 177

CHAPTER 4

Best Practices for Implementing Junos Telemetry Interface

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:

- Reporting interval = Required Export Interval Per Object * Minimum Number of objects reported on each Interval / 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 **JuniperNetworksSensors.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 203 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

Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

You can use subscriber statistics and queue statistics for dynamic interfaces and interface-sets to support remote analytics and monitoring on MX Series routers that operate as a Broadband Network Gateway (BNG).

Before enabling export of subscriber statistics and queue statistics for dynamic interfaces and interface-sets, consider the following limitations:

- On MX Series routers supporting the Modular Port Concentrator 2 (MPC2), a slow internal refresh cycle for queue statistics can occur. This cycle can be lengthy at full line card scale. If the subscription frequency is higher than the internal refresh cycle, exported data may appear stale across reporting intervals.
- The unified in-service software upgrade (ISSU) feature enables you to upgrade your device between two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic. Dynamic interfaces and Interface-sets created prior to ISSU and prior to Junos OS Release 18.4R1 do not support telemetry for subscriber and queue statistics.
- The subscription frequency should be larger than the time to export telemetry. If the volume of data cannot be exported before the next reporting interval, the export continues to completion and the next reporting interval will immediately start. in such instances, continuous streaming results–behavior that may not be wanted.
- Multiple sensors from the dynamic-interfces sub-tree may be subscribed to simultaneously. As streaming of these sensors for the sub-tree is supported by a single Junos component, you should expect the time to export the sensor data for each subscription to extend.
- Juniper advises to enable export only for active queues. To do this, include the **queues** statement at the [**[edit dynamic-profiles *profile-name* telemetryqueue-statistics $junos-interface-name]** or [**[edit dynamic-profiles *profile-name* telemetry queue-statistics $junos-interface-set-name]** hierarchy level. Exporting data for active queues only reduces the amount of data to export for each reporting interval.

Related Documentation

- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Enable Export of Subscriber Statistics and Queue Statistics on page 192

PART 2

Junos Telemetry Interface Plug-ins

CHAPTER 5

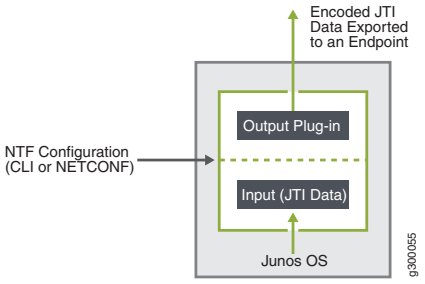
Network Telemetry Framework (NTF) Agent

NTF Agent Overview

Junos OS exposes telemetry data over gRPC and UDP as part of the Junos Telemetry Interface (JTI). One way to stream JTI data into your existing telemetry and analytics infrastructure requires managing an external entity to convert the data into a compatible format. Starting in Junos OS Release 18.4R1, the Network Telemetry Framework (NTF) agent feature provides an on-box solution that allows you to configure and customize to which endpoint (such as IPFIX and Kafka) the JTI data is delivered and in which format (such as AVRO, JSON, and MessagePack) the data is encoded.

NTF agent uses an output plug-in to translate JTI data into a format that is suitable for a particular endpoint. NTF agent subscribes to JTI data with user-defined sensor information. On receiving data, NTF agent uses the output plug-in to encode the data in the format that is required by the endpoint and then exports the translated data to the endpoint (see Figure 5 on page 211). NTF agent can be configured using Junos OS CLI or NETCONF.

Figure 5: NTF Agent Architecture



Related Documentation

- Configuring NTF Agent on page 212

Configuring NTF Agent

To configure a Network Telemetry Framework (NTF) agent instance to send telemetry data to a single endpoint:

1. Create a service agent instance.

```
[edit services analytics agent]
user@host# edit service-agents agent-name
```

2. Configure parameters for the service agent input plug-in. The input plug-in options include **analytics**, **input-ipfix**, and **input-jti-ipfix**. See the inputs configuration statement for a description of the syntax.

NOTE: When you modify the input plug-in configuration of a service agent instance, the associated service agent daemon is restarted.

```
[edit services analytics agent service-agents agent-name]
user@host# edit inputs input-plugin-name parameters key-value-pairs
```

3. Configure parameters for the service agent output plug-in. Parameters are based on the key/value pair requirements of the output plug-in. For each service agent instance, you can configure only one endpoint to which to export data. The output plug-in options include **ouput-ipfix**, **kafka**, and **file**. See the outputs configuration statement for a description of the syntax.

NOTE: When you modify the output plug-in configuration of a service agent instance, the associated service agent daemon is restarted.

```
[edit services analytics agent service-agents agent-name]
user@host# set outputs ouput-plugin-name parameters key-value-pairs
```

4. (Optional) For each service agent instance, you can configure more than one input plug-in to push data to the output plug-in. To illustrate, the basic format of the configuration looks like:

```
[edit services analytics agent service-agents agent1]
  inputs {
  input-plugin1 {
     parameters {
        input-plugin1-key-value-pairs;
     }
  }
  input-plugin2 {
     parameters {
```

```
        input-plugin2-key-value-pairs;
      }
    }
  }
    outputs {
      output-plugin {
        parameters {
          output-plugin-key-value-pairs;
        }
      }
    }
```

5. (Optional) Delete a service agent instance.

```
user@host# delete services analytics agent service-agents agent-name
```

To configure tracing operations for NTF agent:

1. Specify the name of the file to receive the output of the tracing operation. The file is stored in the /var/log/ directory of your device.

```
[edit services analytics agent]
user@host# edit traceoptions filename filename
```

2. Specify the severity level for messages to be logged.

```
[edit services analytics agent]
user@host# edit traceoptions flag {debug | error | info | trace}
```

SHOW COMMANDS and LOG FILES

- Display the running service agent instances of the NTF agent.

```
user@host> show services analytics agent [brief | detail]
```

- You can also view information about service agent instances, such as whether the input and output plug-ins have been initialized, in the service agent log file: /var/log/*agent-name*.log.

Related Documentation

- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- *IPFIX Mediation on the BNG*
- NTF Agent Overview on page 211
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

CHAPTER 6

Open Source Plug-ins

JTI Plug-ins for Open Source Data Collectors

Well-known open source data collectors, such as Telegraf, Fluentd, and Logstash, have a plug-in-based architecture, where Junos Telemetry Interface (JTI) plug-ins can be written to translate JTI data into a format that can be easily understood by the collector. The following table provides links to the public JTI plug-in files for transporting JTI data over UDP and gRPC.

Open Source Data Collector	JTI Plug-ins for Protobuf Encoding over UDP	JTI Plug-ins for OpenConfig key-value Pairs over gRPC
Telegraf	telegraf-jti-plugins	jti_openconfig_telemetry
Fluentd	fluent-plugin-udp-native-sensors	fluent-plugin-grpc-oc-keyvalue
Logstash	logstash-plugin-udp-native-sensors	logstash-plugin-grpc-oc-keyvalue

PART 3

J-Insight Device Monitor

CHAPTER 7

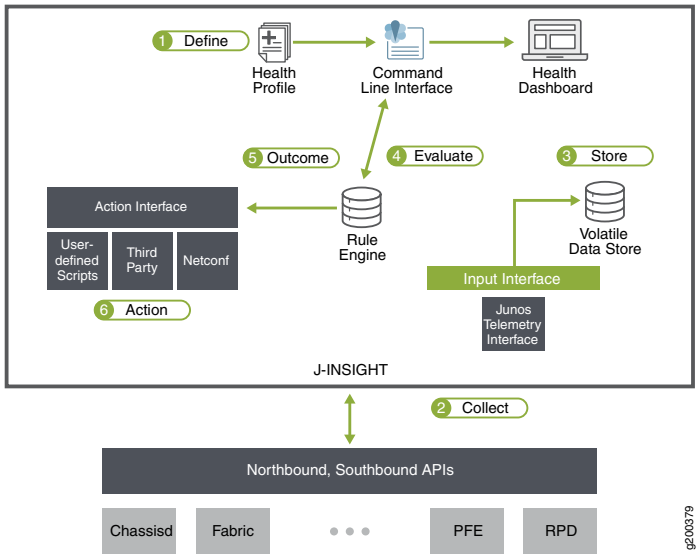
Understanding J-Insight Device Monitor

J-Insight Device Monitor Overview

As networks become increasingly complex, the need to adopt features that simplify the process of monitoring, maintaining, and improving the overall health of your networking devices becomes increasingly critical to delivering services in a more predictable and manageable way.

J-Insight is a data-driven device monitoring solution that provides visibility and insight into the health of a running system. Starting with Junos OS Release 18.2R1, the J-Insight framework facilitates real-time monitoring of system resources for FPC FRUs. It also has been integrated with the existing connectivity error management infrastructure to normalize error detection, monitoring, and reporting. The long-term goal for the architectural design of the J-Insight device monitor is depicted in Figure 6 on page 220.

Figure 6: Long-term High-level Architecture for J-Insight



J-Insight is an on-premise system application that uses the Junos Telemetry Interface to continuously collect data that is reflective of the current state and health of the device component being monitored.

- Understanding How J-Insight Health Monitoring Works on page 220
- Understanding How J-Insight Fault Monitoring Works on page 221

Understanding How J-Insight Health Monitoring Works

Starting in Junos OS Release 18.2R1, J-Insight provides health monitoring capabilities for FPC FRUs on the MX series routers. As part of this initial release, the J-Insight health monitor supports the following process flow (see Figure 6 on page 220):

1. Consumes a pre-defined static health profile. The health profile is not user-configurable through the Junos OS CLI.
2. Using the Junos Telemetry Interface (JTI) framework, subscribes to health KPIs specified in the default health profile. J-Insight health monitor subscribes to JTI sensors using a standard interface. Health monitor subscription and reporting is disabled, by default, and can be enabled through the Junos OS CLI. Starting with Junos OS Release 18.2R1, the following health KPIs are supported for MX-based FPCs:
   - CPU utilization
   - Temperature sensors
   - PFE memory utilization
   - Fabric reachability
3. Collates the JTI data streams collected from various sub-systems.
4. Evaluates the health data against configured thresholds and reports the health status.

Understanding How J-Insight Fault Monitoring Works

Starting with Junos OS Release 18.2R1, J-Insight utilizes the connectivity error management infrastructure to normalize error detection, monitoring, and reporting. Through this infrastructure, J-Insight also provides the capability to define data-driven fault policies. Each module can define error properties by reading a DST/capability file. The fault monitoring capability is available by default in Junos OS and cannot be enabled or disabled through the CLI.

Each error is defined by the following properties:

- **URI**—Error identifier. Each error is uniquely identified with an error ID that is represented as a Uniform Resource Identifier (URI).
- **Error**—Error name.
- **Scope**—Error scope. An error scope provides a level of classification above the error category. Examples of error scope values include: pfe and board.
- **Category**—Error category. An error category categories errors into various subgroups under a specific error scope level. Examples of error category values include: memory, processing, and storage.
- **Details**—Description for the error.
- **Count**—The number of times error instances have occurred.
- **Clear count**—The number of times error instances have been cleared.
- **Support**—Support details for the error type.

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222

J-Insight Device Monitor Basic Configuration

Before you Begin

NOTE: If you're running Junos OS Evolved software, you do not need to perform the procedures in this "Before you Begin" section.

J-Insight requires that your Junos OS device supports the Junos Telemetry Interface (JTI). For information about JTI, see the *Junos Telemetry Interface Feature Guide*. To use J-Insight, you must first complete the following steps:

1. Install the Junos OS Release 18.2R1 or later Junos Network Agent software package. For information on how to install Junos Network Agent, see "Installing the Network Agent Package (Junos Telemetry Interface)" on page 45.

2. Use 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
   [18.2|20180508_0022_builder]
   ```

3. Install the Junos OS Release 18.2R1 or later OpenConfig for Junos OS software package. For information on how to install OpenConfig for Junos OS, see *Installing the OpenConfig Package*.

4. Use the **show version | grep "openconfig"** command to verify that the OpenConfig package was successfully installed.

   ```
   user@host> show version | grep "openconfig"
   JUNOS Openconfig
   [0.0.0|20180503_1001_rbu-builder]
   ```

5. Use the **show agent sensors** command to verify whether or not J-Insight has successfully subscribed to sensors on which it is dependent.

   ```
   user@host> show agent sensors
   .
   .
   .
   Sensor Information :

       Name                                    : sensor_1000
   ```

```
    Resource                                :
/junos/events/event[id='CHASSISD_SNMP_TRAP7']/
    Version                                 : 1.0
    Sensor-id                               : 539528115
    Subscription-ID                         : 1000
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : eventd

    Profile Information :

        Name                                : export_1000
        Reporting-interval                  : 0
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1001
    Resource                                :
/junos/system/cmerror/configuration/
    Version                                 : 1.0
    Sensor-id                               : 539528114
    Subscription-ID                         : 1001
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : PFE

    Profile Information :

        Name                                : export_1001
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1002
    Resource                                : /junos/system/cmerror/counters/

    Version                                 : 1.0
    Sensor-id                               : 539528113
    Subscription-ID                         : 1002
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : PFE

    Profile Information :

        Name                                : export_1002
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1003
    Resource                                : /components/
    Version                                 : 1.0
    Sensor-id                               : 539528112
    Subscription-ID                         : 1003
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : chassisd
```

```
    Profile Information :

        Name                                : export_1003
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1004
    Resource                                :
/junos/services/health-monitor/config/
    Version                                 : 1.0
    Sensor-id                               : 539528119
    Subscription-ID                         : 1004
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : PFE

    Profile Information :

        Name                                : export_1004
        Reporting-interval                  : 7
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1005
    Resource                                :
/junos/services/health-monitor/data/
    Version                                 : 1.0
    Sensor-id                               : 539528118
    Subscription-ID                         : 1005
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : PFE

    Profile Information :

        Name                                : export_1005
        Reporting-interval                  : 7
        Payload-size                        : 5000
        Format                              : GPB
```

J-Insight Health Monitoring

Starting with Junos OS Release 18.2R1, J-Insight supports health monitoring for FPC FRUs on the MX Series routers. The J-Insight health monitor is disabled by default.

- To enable the J-Insight health monitor:

```
user@host# set services jinsightd subscribe health-monitor
```

- To disable the J-Insight health monitor:

```
user@host# delete services jinsightd subscribe health-monitor
```

- To display the J-Insight health monitor results:

```
user@host> show system health-monitor [fpc fpc-slot slot-number]
```

J-Insight Fault Monitoring

Starting with Junos OS Release 18.2R1, J-Insight supports fault monitoring for FPC FRUs on MX Series and PTX Series. Starting with Junos OS Evolved Release 19.1R1, J-Insight fault monitoring support is added for CB, chassis, fan, FPC, FPM, PDU, PIC, PSM, RE and SIB FRUs.

Chassis-level Configuration Commands

The Junos OS resiliency feature provides debugging capabilities in the case of device component failure. You can configure Packet Forwarding Engine (PFE)-related error levels on FRUs such as FPCs. Using the error and fpc error configuration statements, you can set an automatic recovery action for each severity and configure the actions to perform when a specified threshold is reached.

For more information, see the *Chassis-Level Feature Guide*.

Trace Commands

- (Junos OS only) To enable J-Insight trace options for debugging:

```
user@host# set services jinsightd traceoptions flag trace-option
```

- (Junos OS Evolved only) You can view collected J-Insight traces with the show trace **application jinsightd** command, and remove inactive J-Insight tracing sessions with the clear trace **application jinsightd** command.

Clear & Show Commands

- To clear all system errors or a specific error denoted by the error ID Uniform Resource Identifier (URI) for a specific FPC:

```
user@host> clear chassis fpc errors fpc-slot slot-number [ all | error-id error-id- uri]
```

- To display information on alarms that have been triggered by faults:

```
user@host> show chassis alarms
```

- To display summary or detailed information about the active errors based on FRU, error scope, or error category:

```
user@host> show system errors active [[fru slot-number] | [detail [fru slot-number
   [scope error-scope ] [category error-category ]]]
```

- To display a summary of the number of detected errors and recovery actions taken based on severity level:

  ```
  user@host> show system errors count
  ```

- To display information about a detected error based on its error ID URI:

  ```
  user@host> show system errors error-id error-id-uri
  ```

- To display detailed information about the detected errors based on the FRU:

  ```
  user@host> show system errors fru detail [fru slot-number]
  ```

Related Documentation

- J-Insight Device Monitor Overview on page 220

PART 4

Configuration Statements and Operational Commands

CHAPTER 8

Native Sensors Configuration Statements and Operational Commands

export-profile (Junos Telemetry Interface)

Syntax

```
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;
}
```

Hierarchy Level

```
[edit services analytics]
```

Release Information

Statement introduced in Junos OS Release 15.1F3.
payload-size *bytes* option introduced in Junos OS Release 16.1R3.
Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers
loss-priority option introduced in Junos OS Release 17.3R1 for MX Series routers only.
Statement introduced in Junos OS Release 17.3R1 for EX9200 switches and the Routing and Control Board (RCB) on PTX3000 routers.
Statement introduced in Junos OS Release 17.4R1 for virtual MX Series (vMX) routers.

Description

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.

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.

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.

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 236

per-interface-per-member-link

Syntax per-interface-per-member-link (egress *egress-interface* | ingress *ingress-interface*);

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.
Statement introduced in Junos OS Release 18.1R1 on PTX Series routers.

Description Configure sensor-based statistics per interface.

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).

Options **egress *egress-interface***—Enable sensor based statistics on the egress interface.

ingress *ingress-interface*—Enable sensor based statistics on the ingress interface.

NOTE: On PTX Series Routers, the sensor based statistics for SPRING traffic is recorded at the ingress interface only.

Required Privilege Level routing

Related Documentation

- *Understanding Source Packet Routing in Networking (SPRING)*
- *sensor-based-stats*
- per-sid on page 235

per-sid

Syntax

```
per-sid {
  egress;
  ingress;
}
```

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.
egress option introduced in Junos OS Release 19.1R1 on MX Series routers with MPC and MIC interfaces, and PTX series routers.

Description

Configure sensor based statistics per Source Packet Routing in Networking (SPRING) route.

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).

Options

egress—Enable sensor based statistics for IP-MPLS egress accounting. This is supported only for segment routing label IS-IS egress routes at the ingress provider edge (PE) device.

ingress *ingress*—Enable sensor based statistics for per-sid ingress accounting.

Required Privilege Level

routing

Related Documentation

- *Understanding Source Packet Routing in Networking (SPRING)*
- per-interface-per-member-link on page 234
- *sensor-based-stats*

sensor (Junos Telemetry Interface)

Syntax

```
sensor sensor-name {
  export-name export-profile-name;
  polling-interval seconds;
  resource resource-string;
  <resource-filter regular expression>;
  server-name [ streaming-server-names ];
}
```

Hierarchy Level

```
[edit services analytics]
```

Release Information

Statement introduced in Junos OS Release 15.1F3.
Support for MPC7E, MPC8E, and MPC9E on MX Series routers added in Junos OS Release 15.1F5.
Support for FPC1 and FPC2 on PTX Series routers added in Junos OS Release 16.1R3.
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, EX9200 switches, and MX150 routers.
Statement introduced in Junos OS Release 17.4R1 for virtual MX series (vMX) routers.
Statement introduced in Junos OS Release 18.1R1 for MX Series with MS-MICs and MS-MPCs.
Statement introduced in Junos OS Release 18.2R1 for QFX5100, QFX5110, and QFX5200 switches.
Statement introduced in Junos OS Release 18.3R1 for QFX5120-48Y and EX4650 switches.
Statement introduced in Junos OS Release 18.4R1 for EX4600 switches.
Statement introduced in Junos OS Release 18.4R1 for MX480, MX960, MX2010, MX2020, MX2008 and MX-ELM routers.
Statement introduced in Junos OS Release 19.1R1 for MX Series routers operating with MS-MIC and MS-MPC, QFX10002 switches, PTX3000, and PTX10002 routers.
Statement introduced in Junos OS Release 19.1R1 for PTX3000 routers and PTX5000 routers with FPC2.

Description

Configure a Junos Telemetry Interface sensor, which defines the parameters of a system resource to monitor and stream data. 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.

Options

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.

sensor-name—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**.

export-name *export-profile-name*—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.

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.

polling-interval *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)

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 10 on page 239 lists each supported ***resource-identifier-string***, a description of the system resource monitored, and additional configuration information.

You can also use the Telemetry Explorer tool to search for and view information about *resource-identifier-string*.

Table 10: resource statement Options

resource string	Description	Release Information
/junos/events	System events sensor. Starting with Junos OS Release 18.1R1, this sensor corresponds to system log messages (syslog). The sensor must be used with an **export-profile** that has a **reporting-rate** of 0, To subscribe for specific events, you can subscribe for /junos/events/event[id='EVENT_NAME'] where event EVENT_NAME is the event id that you are interested in. Alternatively, you can subscribe to any XPATH Many event names can be found in the messages log file.	Junos OS 18.1R1 and later on all JTI platforms.
/junos/services/ip-tunnel/usage/	Packet forwarding engine packet statistics sensor. The statistics counters are used to report various network element performance metrics in a scalable and efficient way, providing visibility into packet forwarding engine errors and drops. A timestamp indicating when the counters were last reset is included with all the exported data to allow collectors to determine if and when a reset event happened; for example, if the packet forwarding engine hardware restarted. Exported statistics are similar to the output of the operation mode command **show nhdb hw dynamic-ip-tunnels**.	Junos OS 17.4R1 and later on MX Series routers.
/junos/services/label-switched-path/usage/	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. For bypass LSPs, the following are exported: • 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 When the bypass LSP is active, traffic is exported both on the bypass LSP and the ingress (protected) LSP. On MX Series routers only, bidirectional LSPs for ultimate-hop popping (UHP) are also supported. NOTE: You can modify **/junos/services/label-switched-path/usage/** to specify a specific LSP. Add **__instance__/*lsp-name*** to the end of the resource string identifier. For example, to monitor and stream data for LSP statistics for an LSP named	Junos OS Release 15.1F6 and later. Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers. Junos OS Release 17.3 and later on EX9200 and QFX5110 switches. Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches

Table 10: resource statement Options (continued)

resource string	Description	Release Information
	mirror-to-murano-1, enter the following: **/junos/services/label-switched-path/usage/__instance__/mirror-to-murano-1**. If you do not specify a specific LSP name, the system resource monitors and streams data for all LSPs. When you enable a sensor for LSP statistics, you must also configure the sensor-based-stats statement at the **[edit protocols mpls]** hierarchy level. MX Series routers must also operate in enhanced mode. If not enabled by default, configure either the **enhanced-ip** statement or the **enhanced-ethernet** statement at the **[edit chassis network-services]** hierarchy level.	Junos OS Release 19.1R1 and later on QFX10002 Switches and PTX10002 Routers
/junos/services/segment-routing/sid/usage/	Source Packet Routing in Networking (SPRING) sensorSfor transit statistics. SPRING is also known as segment routing. Before statistics can be exported, you must first enable them by including the sensor-based-stats statement at the **[edit protocols isis source-packet-routing]** hierarchy level. For more information, see *Enabling Export of Transit SPRING Statistics*.	Junos OS Release 19.1R1 and later on PTX3000 routers and PTX5000 with FPC2
/junos/services/spu/delegated-rpm/	Delegated Realtime Performance Monitoring (RPM) service sensor. Delegated RPM is a mode where RPM probe generation and measurement calculation are done by MS-MIC and MS-MPC cards. This hardware assistance allows a very high scale of concurrent RPM probes. You can use the resulting data from this sensor to improve network design and optimize traffic engineering. Data can also be used to detect problems in individual devices as well as in the overall network and the traffic carried by it. JTI sensor support for other RPM modes was added in Junos OS Release 18.3R1. This sensor has the following limitations: • Configuring multiple export profiles for the same resource for delegated RPM may not provide expected results. • Multiple sensors for single resource-path (such as delegated RPM) is not supported. • Due to an egress packet-size limitation, history outputs are limited to 5 per RPM test. • The sensor exports one RPM test record per export packet.	Junos OS Release 19.1R1 and later on MX Series routers operating with MS-MIC and MS-MPC
/junos/services/spu/ipsec-vpn		Junos OS 18.1R1 on MX Series with MS-MICs and MS-MPCs

Table 10: resource statement Options (continued)

resource string	Description	Release Information
	UDP-based PIC sensors. Starting with Junos OS Release 18.1R1, this sensor provides visibility for IPSec services on different service complexes and nodes. Exported data is defined using an IP address and a UDP port. When an export interval expires, the most recent statistics collected by the sensors are gathered, placed in the payload of a UDP packet, and forwarded to a collector. A timestamp indicating when counters are read is included with the exported data to allow collectors to collate data. The timestamp also can determine if and when an event happened, such as a PIC hardware restart or if counters were cleared by means of the CLI.	
/junos/services/spu/servicesets	Sensor to export service set statistics. These sensors provide visibility for services on different service complexes and nodes (for example, IPSec services). Exported data is defined using an IP address and a UDP port. When an export interval expires, the most recent statistics collected by the sensors are gathered, placed in the payload of a UDP packet, and forwarded to a collector. A timestamp indicating when counters are read is included with the exported data to allow collectors to collate data. The timestamp also can determine if and when an event happened, such as a PIC hardware restart or if counters were cleared by means of the CLI.	Junos OS 18.2R1 on MX Series with MS-MICs and MS-MPCs
/junos/services/spu/sessions	Sensor to export session statistics. These sensors provide visibility for services on different service complexes and nodes (for example, IPSec services). Exported data is defined using an IP address and a UDP port. When an export interval expires, the most recent statistics collected by the sensors are gathered, placed in the payload of a UDP packet, and forwarded to a collector. A timestamp indicating when counters are read is included with the exported data to allow collectors to collate data. The timestamp also can determine if and when an event happened, such as a PIC hardware restart or if counters were cleared by means of the CLI.	Junos OS 18.2R1 on MX Series with MS-MICs and MS-MPCs
/junos/system/linecard/node-slicing/af-fab-stats/	Sensor to export abstracted fabric (AF) interface specific load-balancing and fabric queue statistics. This sensor is only supported for a node virtualization configuration on MX series routers with an AF Interface as the connecting link between guest network functions (GNFs). The sensor also reports aggregated statistics across all AF interfaces hosted on a source packet forwarding engine of local GNFs along with the fabric statistics for all traffic ingressing from and egressing to the fabric from that the packet forwarding engine.	Junos OS Release 18.4R1 and later on MX480, MX960, MX2008, MX2010, MX2020, and MX-ELM routers

Table 10: resource statement Options (continued)

resource string	Description	Release Information
/junos/system/linecard/cpu/memory/	Packet Forwarding Engine sensor for CPU memory.	Junos OS Release 16.1R3 and later. Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers. Junos OS Release 17.3R1 and later on EX9200 and QFX5110 switches. Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches Junos OS Release 19.1R1 and later on QFX10002 Switches and PTX10002 Routers
/junos/system/linecard/firewall/	Packet Forwarding Engine sensor for firewall filter counters and policer counters. Each line card reports counters separately. 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. Firewall counters are exported even if the interface to which the firewall filer is attached is down.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 15.1F5 and later. Junos OS Release 17.2R1 and later on QFX10000 switches. Junos OS Release 17.3R1 and later on PTX1000 routers and EX9200 switches and QFX5110 switches.. Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches
/junos/system/linecard/interface/	Packet Forwarding Engine sensor for physical interface traffic. 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. For Aggregated Ethernet interfaces, statistics are exported for the member physical interfaces. You must aggregate the counters at the destination server, or collector. If a physical interface is administratively down or operationally down, interface counters are not exported. Issuing an operational **clear** command, such as **clear interfaces statistics all**, does not reset statistics exported by the line card.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 15.1F3 and later on PTX Series routers only. Support introduced for MX Series routers in Junos OS Release 15.1F5. Junos OS Release 17.2R1 and later on QFX10000 switches and PTX1000 routers. Junos OS Release 17.3R1 and later on EX9200 switches, QFX5110 switches and MX150 routers. Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches Junos OS Release 19.1R1 and later on QFX10002 Switches and PTX10002 Routers
/junos/system/linecard/interface/logical/usage/	Packet Forwarding Engine sensor for logical interface statistics. NOTE: If a logical interface is operationally down, interface statistics continue to be exported. Issuing an operational **clear** command, such as **clear interfaces statistics all**, does not reset statistics exported by the line card. NOTE: Locally injected packets from the Routing Engine are not exported.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 15.1F5 and later. Junos OS Release 17.2R1 and later on QFX10000 switches. Junos OS Release 17.3R1 and later on EX9200 and QFX5110 switches Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches
/junos/system/linecard/interface/traffic/	Packet Forwarding Engine sensor for physical interface traffic. Exports all fields from **/junos/system/linecard/interface/** *except* queue statistics This additional sensor can reduce the reap time for non-queue data for platforms supporting VoQ architecture. To export traffic and queue data for physical interfaces, use **/junos/system/linecard/interface/**. To export queue fields only, use **/junos/system/linecard/interface/queue/**.	Junos OS Release 18.3R1 and later on PTX Series and ACX Series routers and EX Series, MX Series, and QFX Series switches.
/junos/system/linecard/interface/queue/	Packet Forwarding Engine sensor for physical interface traffic. Exports all queue fields from **/junos/system/linecard/interface/**. To export traffic and queue data for physical interfaces, use **/junos/system/linecard/interface/**. To export traffic fields only, use **/junos/system/linecard/interface/traffic/**.	Junos OS Release 18.3R1 and later on PTX Series and ACX Series routers and EX Series, MX Series, and QFX Series switches.
/junos/system/linecard/npu/memory/	Packet Forwarding Engine sensor for network processing unit (NPU) memory.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 16.1R3 and later. Junos OS Release 17.2R1 and later on QFX10000 switches. Junos OS Release 17.3R1 and later on EX9200 switches. Junos OS Release 19.1R1 and later on PTX10002 Routers.
/junos/system/linecard/npu/utilization/	Packet Forwarding Engine sensor for NPU processor utilization.	Junos OS Release 16.1R3 and later. Junos OS Release 17.2R1 and later on QFX10000 switches. Junos OS Release 17.3R1 and later on EX9200 switches. Junos OS Release 19.1R1 and later on PTX10002 Routers.
/junos/npu-memory/	Sensor that exports both NPU memory statistics from the Packet Forwarding Engine and flow-label statistics from the Routing Engine. To export only flow-label statistics, include the **junos/npu-memory/flabel-memory/** resource string.	Junos OS Release 16.1R3 and later on PTX Series routers only. NOTE: Junos OS Release 17.2R1 and later on PTX1000 routers.
/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.	Junos OS Release 16.1R3 and later on MX series and PTX series routers only. Junos OS Release and later on EX9200 switches, PTX1000 routers, and MX150 routers.
/junos/system/linecard/optics/	Packet Forwarding Engine sensor for various optical performance metrics, such as transmit and receive power levels.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 17.1R1 and later. Junos OS Release and later 17.2R1 on QFX10000 switches. Junos OS Release 17.3R1 and later on EX9200 switches and PTX1000 routers.

/junos/system/linecard/qmon/	Sensor for queue depth statistics for ingress and egress queue traffic. Statistics are exported directly from the line card.
	
	This sensor only supports single -streaming. Configuring this sensor to stream to multiple servers is not supported. If multiple servers are configured, no data is sent to any of the configured servers.
	
	The following example shows a configuration for single-streaming that will send data:
	
	`sensor qmon {`
	`server-name TEMP;`
	`export-name export-common;`
	`resource /junos/system/linecard/qmon/;`
	`}`
	
	The following example shows a multiple-server configuration that will not send data:
	
	`sensor qmon {`
	`server-name TEMP;`
	`server-name digi1;`
	`server-name digi2;`
	`export-name export-common;`
	`resource /junos/system/linecard/qmon/;`
	`}`
	
	NOTE: Issuing an operational **clear** command, such as **clear interfaces statistics all**, does not reset the statistics exported by the line card.
		Junos OS Release 17.1R1 and later on MX Series routers on MPC7E, MPC8E, and MPC9E only.
		
		Junos OS 17.3R1 and later on EX9200 switches.
		
		NOTE: virtual MX Series (vMX) routers are not supported.

/junos/system/linecard/qmon-sw/	Sensor for congestion and latency monitoring statistics.	

Table 10: resource statement Options (continued)

resource string	Description	Release Information
		Junos OS Release 18.2R1 and later on QFX5100, QFX5110, and QFX5200 Switches Junos OS Release 18.3R1 and later on QFX5120-48Y and EX4650 Switches Junos OS Release 18.4R1 and later on EX4600 Switches
/junos/system/linecard/fabric/	Sensor for fabric statistics. The following types of statistics can be exported: • Fabric statistics for Packet Forwarding Engine pairs (**resource-filter** option is not supported) • FPC fabric statistics • Control Board and Switch Fabric Board fabric statistics.	Junos OS Release 17.2R1 and later on MX Series routers only. Junos OS Release 17.3R1 and later on EX9200 switches. NOTE: virtual MX Series (vMX) routers are not supported.
/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 PTX Series routers Junos OS Evolved Release 19.1R1 on PTX10003 routers and QFX10003 switches
/junos/services/segment-routing/interface/ingress/usage/ /junos/services/segment-routing/interface/egress/usage/ /junos/services/segment-routing/sid/usage/	Sensors for aggregate segment routing traffic with IS-IS. The first path exports inbound traffic. The second path exports outbound traffic. The third path exports inbound segment routing traffic for each segment identifier. 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.	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_testl*

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 Level

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

Release History Table

Release	Description
19.1R1	Statement introduced in Junos OS Release 19.1R1 for PTX3000 routers and PTX5000 routers with FPC2.

Related Documentation

- export-profile on page 230

sensor-based-stats (Junos Telemetry Interface)

Syntax

```
sensor-based-stats;
```

Hierarchy Level

```
[edit protocols mpls]
[edit protocols isis source-packet-routing]
```

Syntax

```
sensor-based stats {
  per-interface-per-member-link (ingress interface-name | egress interface-name);
  per-sid ingress interface-name;
}
```

Hierarchy Level

```
[edit protocols isis source-packet-routing]
```

Release Information

Statement introduced in Junos OS Release 15.1F6.
Statement introduced in Junos OS Release 17.2R1 for QFX10000 switches and PTX1000 routers.
Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.
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.
Statement supported in Junos OS Evolved Release 19.1R1 on PTX10003 routers and QFX 10003 switches.

Description

For the MPLS hierarchy, enable the collection of LSP statstics 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 ***sensor-name*****]** hierarchy level.

For additional information about configuring an LSP statistics sensor to stream data through gRPC, see "Guidelines for gRPC Sensors (Junos Telemetry Interface)" on page 51.

For the IS-IS hierarchy, enable the collection of aggregate segment routing statistics.

NOTE: Only MX Series routers, PTX3000, and PTX5000 routers support this hierarchy.

Options

The remaining options are explained separately.

Required Privilege Level

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

Related Documentation

- Understanding the Junos Telemetry Interface Export Format of Collected Data on page 8

source-packet-routing

Syntax

```
source-packet-routing {
  telemetry {
    statistics {
      no-transit;
      no-ingress;
      }
    }
  }
```

Hierarchy Level

```
[edit protocols]
```

Release Information Statement introduced in Junos OS Release 18.3R1 for MX Series and PTX Series routers.

Description Enable BGP and statically configured Segment Routing Traffic Engineering (SR-TE) traffic statistics sensor support for Junos Telemetry Interface (JTI).

Export JTI statistics using either gRPC streaming or UDP native sensors. The following resource paths are supported. For UDP native sensors:

- /junos/services/segment-routing/traffic-engineering/ingress/usage/
- /junos/services/segment-routing/traffic-engineering/transit/usage

For gRPC streaming:

- /mpls/signaling-protocols/segment-routing/

For exporting statistics using UDP native sensors, configure parameters at the [edit services analytics] hierarchy level. To provision the sensor to export data through gRPC streaming, use the telemetrySubscribe RPC to specify telemetry parameters.

Default Disabled.

Options

statistics—Create sensors for both the SR-TE policy nexthop and the binding SID that are installed in the forwarding plane. For the SR-TE policy nexthop, the sensors collect traffic statistics steered by all routes that use the SR-TE policy as a nexthop. For the binding SID, the sensors collect statistics on labeled traffic that is steered by the binding-SID route.

no-transit—Enable sensors only for SR-TE policy nexthops. The sensor will collect statistics on all steering routes that use the SR-TE policy as a nexthop.

no-ingress—Enable sensors only for Binding-SID transit routes.

Required Privilege Level routing

Related Documentation

- Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32
- Configure a Telemetry Sensor in Junos on page 177
- **sensor (Junos Telemetry Interface) on page 236**
- *statistics*
- *telemetry*

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 Level

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

Related Documentation

- export-profile (Junos Telemetry Interface) on page 230

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 230
- sensor on page 236
- streaming-server on page 254

List of Sample Output
show agent sensors (firewall filter sensor) on page 257
show agent sensors (CPU memory sensor) on page 258
show agent sensors (packet forwarding engine statistics) on page 259
show agent sensors (QFX10008 or QFX10016 switches with Junos OS Release 17.3R1 and later) on page 259
show agent sensors (Junos OS Evolved Release 19.1R1 and later) on page 260

Output Fields Table 11 on page 256 lists the output fields for the **show agent sensors** command. Output fields are listed in the approximate order in which they appear.

Table 11: show agent sensors Output Fields

Field Name	Field Description
Sensor Information	Information about sensors configured to monitor system resources and stream data.

Table 11: show agent sensors Output Fields (continued)

Field Name	Field Description
Name	Name of configured sensor. NOTE: Junos OS Evolved Release 19.1R1 and later does not show output for generated child sensors.
Resource	Resource string used to configure and identify the system resource enabled to monitor and stream data.
Sensor-id	Numerical identifier of the sensor.
Server Information	Information about servers configured to collect sensor data.
Name	Name of server.
Scope-id	Numerical identifier of a scope.
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
```

show agent sensors (packet forwarding engine statistics)

```
user@host> show agent sensors
Sensor Information :

    Name                                    : packet_stats
    Resource                                : /junos/system/linecard/packet/usage/

    Version                                 : 1.0
    Sensor-id                               : 3699
    Subscription-ID                         : 562949953425011
    Parent-Sensor-Name                      : Not applicable
    Component(s)                            : PFE

    Server Information :
        Name                                : s1
        Scope-id                            : 0
        Remote-Address                      : 10.1.1.2
        Remote-port                         : 1000
        Transport-protocol                  : UDP

    Profile Information :
        Name                                : ep1
        Reporting-interval                  : 1
        Payload-size                        : 5000
        Address                             : 10.1.1.1
        Port                                : 1000
        Timestamp                           : 1
        Format                              : GPB
        DSCP                                : 255
        Forwarding-class                    : 255
```

show agent sensors (QFX10008 or QFX10016 switches with Junos OS Release 17.3R1 and later)

```
user@host> show agent sensors
 Sensor Information :

     Name                                   : sensor_1000
     Resource                               : /interfaces/interface/subinterfaces/

     Version                                : 1.0
     Sensor-id                              : 539528115
     Subscription-ID                        : 1000
     Parent-Sensor-Name                     : Not applicable
     Component(s)                           : PFE,mib2d,xmlproxyd

     Profile Information :

         Name                               : export_1000
         Reporting-interval                 : 6
         Payload-size                       : 5000
         Format                             : GPB

 Sensor Information :

     Name                                   : sensor_1000_1_1
     Resource                               :
/junos/system/linecard/interface/logical/usage/
```

```
    Version                                 : 1.1
    Sensor-id                               : 3139259737
    Subscription-ID                         : 1000
    Parent-Sensor-Name                      : sensor_1000
    Component(s)                            : PFE

    Profile Information :

        Name                                : export_1000
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1000_2_1
    Resource                               : /interfaces/interface/subinterfaces/

    Version                                 : 1.0
    Sensor-id                               : 3139256665
    Subscription-ID                         : 1000
    Parent-Sensor-Name                      : sensor_1000
    Component(s)                            : mib2d

    Profile Information :

        Name                                : export_1000
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB

Sensor Information :

    Name                                    : sensor_1000_4_1
    Resource                               : /interfaces/interface/subinterfaces/

    Version                                 : 1.0
    Sensor-id                               : 3139262809
    Subscription-ID                         : 1000
    Parent-Sensor-Name                      : sensor_1000
    Component(s)                            : xmlproxyd

    Profile Information :

        Name                                : export_1000
        Reporting-interval                  : 6
        Payload-size                        : 5000
        Format                              : GPB
```

show agent sensors (Junos OS Evolved Release 19.1R1 and later)

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

```
Sensor Information :

    Name                                    : sensor_1000
    Resource                                :
/interfaces/interface[name='re0:mgmt-0']/
    Version                                 : 1.0
    Sensor-id                               : 562949953421313
```

```
Subscription-ID                         : 1000
Component(s)                            : mib2d, mgmt-ethd

Profile Information :

    Name                                : export_1000
    Reporting-interval                  : 2
    Payload-size                        : 5000
    Address                             : 0.0.0.0
    Port                                : 1000
    Timestamp                           : ntp
    Format                              : GPB
    DSCP                                : 0
    Forwarding-class                    : 0
```

CHAPTER 9

gRPC Services Configuration Statements and Operational Commands

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]>
<snmp>
```

Release Information

Command introduced in Junos OS Release 16.1R1 on MX Series and T Series routers.
Command introduced in Junos OS Release 17.1R1 on EX Series and QFX Series switches and PTX Series routers.
Command introduced in Junos OS Release 17.3R1 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.
proxy-xml option introduced in Junos OS Release 17.3R1 on MX Series and PTX Series routers.
Command introduced in Junos OS Release 18.1R1 on ACX Series routers.
snmp option introduced in Junos OS Release 18.3R1.

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.

NOTE: Starting in Junos OS Release 18.3R1, adding, deleting, or updating YANG packages in configuration mode with the **run** command is not supported.

NOTE: Devices that use the ephemeral configuration database will delete all ephemeral configuration data in the process of rebuilding the schema.

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.

snmp —List of paths for one or more YANG modules to copy to a predefined location and convert it to JSON format. Later snmpd parses this JSON file and builds its internal database. Requires the **package *package-name*** option.

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*
- Configure a Telemetry Sensor in Junos on page 177
- request system yang update on page 269
- *show system yang package*
- *Customized SNMP MIBs for Syslog Traps*

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 16.1R1 on MX Series and T Series routers.
Command introduced in Junos OS Release 17.1R1 on EX Series and QFX Series switches and PTX Series routers.
Command introduced in Junos OS Release 17.3R1 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.
Command introduced in Junos OS Release 18.1R1 on ACX Series routers.

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.

NOTE: Starting in Junos OS Release 18.3R1, adding, deleting, or updating YANG packages in configuration mode with the **run** command is not supported.

NOTE: Devices that use the ephemeral configuration database will delete all ephemeral configuration data in the process of rebuilding the schema.

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 264
- *show system yang package*

Sample Output

request system yang delete

```
user@host> request system yang delete p1
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 package-name action-script [scripts]
  deviation-module [modules] module [modules] proxy-xml [file-path-names]
  translation-script [scripts]
```

Release Information

Command introduced in Junos OS Release 16.1R1 on MX Series and T Series routers.
Command introduced in Junos OS Release 17.1R1 on EX Series and QFX Series switches and PTX Series routers.
Command introduced in Junos OS Release 17.3R1 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.
proxy-xml option introduced in Junos OS Release 17.3R1 on MX Series and PTX Series routers.
Command introduced in Junos OS Release 18.1R1 on ACX Series routers.

Description

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.

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.

NOTE: Starting in Junos OS Release 18.3R1, adding, deleting, or updating YANG packages in configuration mode with the run command is not supported.

NOTE: Devices that use the ephemeral configuration database will delete all ephemeral configuration data in the process of rebuilding the schema.

Options

package-name—Name of the YANG package to update.

action-script [*scripts*]—List of paths for one or more action scripts to add to or update in the package.

deviation-module [*modules*]—List of paths for one or more deviation modules to add to or update in the package.

module [*modules*]—List of paths for one or more YANG modules to add to or update in the package.

proxy-xml [*file-path-names*]—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.

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

Required Privilege Level maintenance

Related Documentation

- *Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS*
- Configure a Telemetry Sensor in Junos on page 177
- request system yang add on page 264
- *show system yang package*

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

request system yang validate action-script [*scripts*] module [*modules*] proxy-xml module [*modules*] translation-script [*scripts*]

Release Information

Command introduced in Junos OS Release 16.1R1 on MX Series and T Series routers.
Command introduced in Junos OS Release 17.1R1 on EX Series and QFX Series switches and PTX Series routers.
Command introduced in Junos OS Release 17.3R1 on SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.
proxy-xml option introduced in Junos OS Release 17.3R1 on MX Series and PTX Series routers.
Command introduced in Junos OS Release 18.1R1 on ACX Series routers.

Description

Validate the syntax of one or more YANG modules, translation scripts, or action scripts.

Options

action-script *scripts*—List of paths for one or more action scripts to validate.

module *modules*—List of paths for one or more YANG modules to validate.

proxy-xml module *modules*—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.

translation-script *scripts*—List of paths for one or more translation scripts to validate.

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*
- Configure a Telemetry Sensor in Junos on page 177

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
```

source-packet-routing

Syntax

```
source-packet-routing {
  telemetry {
    statistics {
      no-transit;
      no-ingress;
      }
    }
  }
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 18.3R1 for MX Series and PTX Series routers.

Description Enable BGP and statically configured Segment Routing Traffic Engineering (SR-TE) traffic statistics sensor support for Junos Telemetry Interface (JTI).

Export JTI statistics using either gRPC streaming or UDP native sensors. The following resource paths are supported. For UDP native sensors:

- /junos/services/segment-routing/traffic-engineering/ingress/usage/
- /junos/services/segment-routing/traffic-engineering/transit/usage

For gRPC streaming:

- /mpls/signaling-protocols/segment-routing/

For exporting statistics using UDP native sensors, configure parameters at the [edit services analytics] hierarchy level. To provision the sensor to export data through gRPC streaming, use the telemetrySubscribe RPC to specify telemetry parameters.

Default Disabled.

Options

statistics—Create sensors for both the SR-TE policy nexthop and the binding SID that are installed in the forwarding plane. For the SR-TE policy nexthop, the sensors collect traffic statistics steered by all routes that use the SR-TE policy as a nexthop. For the binding SID, the sensors collect statistics on labeled traffic that is steered by the binding-SID route.

no-transit—Enable sensors only for SR-TE policy nexthops. The sensor will collect statistics on all steering routes that use the SR-TE policy as a nexthop.

no-ingress—Enable sensors only for Binding-SID transit routes.

Required Privilege Level routing

Related Documentation

- Understanding OpenConfig and gRPC on Junos Telemetry Interface on page 32
- Configure a Telemetry Sensor in Junos on page 177
- **sensor (Junos Telemetry Interface) on page 236**
- *statistics*
- *telemetry*

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-verfiy;
    }
  }
  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-verfiy—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-authorirty 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 32767.

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

- *grpc*
- *JET Service Process Overview*
- *Configuring Request-Response Service for JET Applications*

telemetry

Syntax

```
telemetry {
  subscriber-statistics;
  queue-statistics {
    interface $junos-interface-name {
      refresh rate;
      queues queue-set;
    }
    interface-set $junos-interface-set-name {
      refresh rate;
      queues queue-set;
    }
  }
}
```

Hierarchy Level

[edit dynamic-profiles *profile-name*]

Release Information

Statement introduced in Junos OS 18.4.

Description

Enable telemetry collection of subscriber statistics and queue statistics.

Include the **subscriber-statistics** statement to enable the export of subscriber statistics through telemetry. When this statement is configured, you must also include the **actual-transit-statistics** statement at the [**edit dynamic-profiles** ***profile-name*** **interfaces** ***interface-name*** **unit** ***unit-name***} hierarchy level to enable subscriber-statistics.

Include the **queue-statistics** statement to instruct the statistics infrastructure to collect queue statistics for dynamic interfaces or interface-sets queue-statistics and enable export via Junos Telelemetry Interface (JTI).

The profile variable **$junos-interface-name"** and **"$junos-interface-set-name"** are generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time. Using these derived variables is a convenient way to configure telemetry behavior for the interface or interface-set without the need to mimic the specific configuration in the interfaces stanza.

After telemetry for these statistics is enabled, they are eligible for export through a collector subscription.

For information about subscribing to the statistics through an external collector, see "Configure a Telemetry Sensor in Junos" on page 177. For information about supported sensors for subscriber statistics and queue statistics, see "Guidelines for gRPC Sensors (Junos Telemetry Interface)" on page 51.

Options

subscriber-statistics—Enable the export of interface meta-data and export of subscriber accurate statistics. When this statement is configured, you must also include the

actual-transit-statistics statement at the [**edit dynamic-profiles** *profile-name* **interfaces** *interface-name* **unit** *unit-name*] hierarchy level.

queue statistics interface "$junos-interface-name"**—Enable the export of interface meta-data and interface queue statistics. The profile variable **$junos-interface-name" is generated from the corresponding device, unit and interface elements in the interfaces stanza at profile instantiation time.

queue statistics interface-set "$junos-interface-set-name"**—Enable the export of interface-set meta-data and interface-set queue statistics. The profile variable **"$junos-interface-set-name" is generated from the corresponding device, unit and interface-set elements in the interfaces stanza at profile instantiation time.

refresh *rate*—Override the default internal queue statistics collection interval. If dynamic interfaces and interface-sets are created as a result of multiple dynamic profiles, each with their own refresh intervals, the smallest interval for each object type (interface or interface-set) is used to poll queue statistics for that object type. The default is 900 seconds.

Range: 300 seconds (5 minutes) to 86,400 seconds (24 hours)

queue "*queue-set*"—Specify the set of queues for which queue-statistics will be exported. The queue set is a comma delimited string of integers. The default is all queues (**0,1,2,3,4,5,6,7**) are eligible for export.

Range: 0 to 7

Required Privilege Level

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

Related Documentation

- Enable Export of Subscriber Statistics and Queue Statistics on page 192
- Understanding Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 190
- Guidelines for Exporting Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets on page 193
- Guidelines for gRPC Sensors (Junos Telemetry Interface) on page 51

CHAPTER 10

Network Telemetry Framework (NTF) Configuration Statements and Operational Commands

agent (Analytics)

Syntax

```
agent {
  service-agents {
    agent-name {
      inputs {
        analytics {
          parameters {
            generate-tags value;
            sample-frequency value;
            sensors file-path;
          }
        }
        input-ipfix {
          parameters {
            maximum-connections number;
            tcp-port port-number;
            vrf-name name;
          }
        }
        input-jti-ipfix {
          parameters {
            record-group group-name {
              record ipfix-record-name;
              reporting-interval seconds;
            }
          }
        }
      }
      outputs {
        file {
          parameters {
            path file-path;
          }
        }
        kafka {
          parameters {
            server ip-address;
            topic topic-name;
            encoding encoding-type;
          }
        }
        output-ipfix {
          parameters {
            collector-address ip-address;
            collector-ca-certificate file-path;
            collector-certificate file-path;
            collector-certificate-key file-path;
            collector-connection-retry-interval seconds;
            collector-tcp-port port-number;
            collector-vrf-name vrf-name;
          }
        }
```

```
      }
    }
  }
  traceoptions {
    filename filename;
    flag (debug | error | info | trace);
  }
}
```

Hierarchy Level

[edit services analytics]

Release Information

Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.

Description

Configure the Network Telemetry Framework (NTF) agent and corresponding service agents that use input and output plug-ins to collect, transform, and forward network telemetry data.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

system

Related Documentation

- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- Configuring NTF Agent on page 212
- *IPFIX Mediation on the BNG*
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

analytics

List of Syntax

Syntax (EX Series and QFX Series) on page 282
Syntax (MX Series & PTX Series) on page 285

Syntax (EX Series and QFX Series)

Junos OS Release 13.2X51-D15 and later:

```
analytics {
  collector {
    local {
      file filename {
        size size;
        files number;
      }
    }
    address ip-address {
      port number {
        transport protocol {
          export-profile profile-name;
        }
      }
    }
  }
  export-profiles {
    profile-name {
      interface {
        information;
        statistics {
          queue;
          traffic;
        }
        status {
          link;
          queue;
          traffic;
        }
      }
      stream-format format;
      system {
        information;
        status {
          queue;
          traffic;
        }
      }
    }
  }
  resource {
    interfaces {
      interface-name {
        resource-profile name;
      }
    }
```

```
    system {
      polling-interval {
        queue-monitoring interval;
        traffic-monitoring interval;
      }
      resource-profile name;
    }
  }
  resource-profiles {
    profile-name {
      depth-threshold {
        high number;
        low number;
      }
      latency-threshold {
        high number;
        low number;
      }
      no-queue-monitoring;
      no-traffic-monitoring;
      queue-monitoring;
      traffic-monitoring;
    }
  }
  traceoptions {
    file filename {
      files number;
      size size;
    }
  }
}
```

Junos OS Release 13.2X50-D15 and 13.2X51-D10 only:

```
analytics {
  interfaces {
    all {
      depth-threshold high number low number;
      latency-threshold high number low number;
      queue-statistics;
      no-queue-statistics;
      traffic-statistics;
      no-traffic-statistics;
    }
    interface-name {
      depth-threshold high number low number;
      latency-threshold high number low number;
      queue-statistics;
      no-queue-statistics;
      traffic-statistics;
      no-traffic-statistics;
    }
  }
  queue-statistics {
    file filename {
      files number-of-files;
      size size;
    }
    interval interval;
  }
  streaming-servers {
    address ip-address {
      port number {
        stream-format format;
        stream-type type
      }
    }
  }
  traceoptions {
    file filename {
      files number;
      size size;
    }
  }
  traffic-statistics {
    file filename {
      files number-of-files;
      size size;
    }
    interval interval;
  }
}
```

Syntax (MX Series & PTX Series)

```
analytics {
  agent {
    service-agents {
      agent-name {
        inputs {
          analytics {
            parameters {
              generate-tags value;
              sample-frequency value;
              sensors file-path;
            }
          }
          input-ipfix {
            parameters {
              maximum-connections number;
              tcp-port port-number;
              vrf-name name;
            }
          }
          input-jti-ipfix {
            parameters {
              record-group group-name {
                record ipfix-record-name;
                reporting-interval seconds;
              }
            }
          }
        }
        outputs {
          file {
            parameters {
              path file-path;
            }
          }
          kafka {
            parameters {
              server ip-address;
              topic topic-name;
              encoding encoding-type;
            }
          }
          output-ipfix {
            parameters {
              collector-address ip-address;
              collector-ca-certificate file-path;
              collector-certificate file-path;
              collector-certificate-key file-path;
              collector-connection-retry-interval seconds;
              collector-tcp-port port-number;
              collector-vrf-name vrf-name;
            }
          }
        }
      }
    }
  }
```

```
    traceoptions {
      filename filename;
      flag (debug | error | info | trace);
    }
  }
}
```

Hierarchy Level

[edit services]

Release Information

Statement introduced in Junos OS Release 13.2 on QFX Series switches.
Statement introduced in Junos OS Release 13.2X51-D25 on EX Series switches.
Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.

Description

Configure the network analytics feature that includes monitoring for traffic and queue statistics. The network analytics processes running on the Packet Forwarding Engine and Routing Engine collect and analyze the data, and generate reports that may be saved in log files or sent as streaming data to remote servers.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

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

Related Documentation

- *Network Analytics Overview*
- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- Configuring NTF Agent on page 212
- *IPFIX Mediation on the BNG*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

inputs (Analytics)

Syntax

```
inputs {
  analytics {
    parameters {
      generate-tags value;
      sample-frequency value;
      sensors path;
    }
  }
  input-ipfix {
    parameters {
      maximum-connections number;
      tcp-port port-number;
      vrf-name name;
    }
  }
  input-jti-ipfix {
    parameters {
      record-group group-name {
        record ipfix-record-name;
        reporting-interval seconds;
      }
    }
  }
}
```

Hierarchy Level

[edit services analytics agent service-agents *agent-name*]

Release Information

Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.
input-jti-ipfix option added in Junos OS Release 18.4R1 on MX Series routers.
analytics option added in Junos OS Release 18.4R1 on MX Series and PTX Series routers.

Description

Configure parameters for a Network Telemetry Framework (NTF) service agent input plug-in. For each service agent instance, you can configure more than one input plug-in to push data to the output plug-in.

NOTE: When you modify the input plug-in configuration of a service agent instance, the associated service agent daemon is restarted.

Options

analytics parameters—Configure parameters to collect data from Junos Telemetry Interface (JTI) sensors.

generate-tags *value*—(Optional) Enable tag generation.

Default: Enabled

sample-frequency *value*—Specify the frequency interval (in milliseconds) at which the JTI sensor generates data to export to the data collector.

Default: 5000 milliseconds

sensors *file-path*—Specify the resource string associated with the JTI sensor for collecting JTI data from a specific resource. The format is a file path and must be entered exactly. For a list of available JTI resource string options, see the **sensor** configuration statement and "Guidelines for gRPC Sensors (Junos Telemetry Interface)" on page 51 documentation.

input-ipfix parameters—Configure parameters for the IPFIX mediation service agent to gather and consolidate IPFIX records from downstream devices.

NOTE: Any change you make to an existing **input-ipfix** plug-in configuration restarts the IPFIX service agent daemon to apply the changes.

NOTE: Although each of the parameters has a default value, you must configure at least one of the parameters to enable the plug-in. If you configure only one parameter and want to use the default value, you must specify that value.

maximum-connections *number*—(Optional) Maximum number of TCP connections that the IPFIX mediator can support.

Range: 1 through 500

Default: 100

tcp-port *port-number*—(Optional) TCP port on the IPFIX mediator that receives TCP packets; the listening port.

Default: 4739

vrf-name *name*—(Optional) Name of the VRF (routing instance) in which IPFIX packets are accepted.

Default: default

input-jti-ipfix parameters—Configure parameters for the IPFIX mediation service agent to collect and report local sensor data from the BNG configured as an IPFIX mediator. For each group of records, the plug-in subscribes to the specific sensor data sets associated with each record.

When you remove a record group from the configuration, the sensor sets for the member records are unsubscribed. The template IDs for the associated IPFIX records are returned to the pool for re-use.

record *ipfix-record-name*—One of the following individual IPFIX records associated with a nonconfigurable set of local sensor data. See *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector* for the sensors collected by each record.

address-pool-utilization	port-statistics
chassis-inventory	resource-utilization
chassis-power	subscriber-statistics
dhcpv4-server-stats	thermal
interface-metadata	uptime

record-group *group-name*—Name of a group of IPFIX records that subscribes to the sensor data sets associated with the individual records that comprise the record group. You can configure a maximum of 10 record groups.

reporting-interval *seconds*—(Optional) Interval in seconds between reports for the subscribed sensor data. The interval applies to all records (and all sensor sets) in the record group.

Range: 60 through 86,400 seconds

Default: 900 seconds

Required Privilege Level system

Related Documentation

- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- Configuring NTF Agent on page 212
- *IPFIX Mediation on the BNG*
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

outputs (Analytics)

Syntax

```
outputs {
  file {
    parameters {
      path file-path;
    }
  }
  kafka {
    parameters {
      server ip-address;
      topic topic-name;
      encoding encoding-type;
    }
  }
  output-ipfix {
    parameters {
      collector-address ip-address;
      collector-ca-certificate file-path;
      collector-certificate file-path;
      collector-certificate-key file-path;
      collector-connection-retry-interval seconds;
      collector-tcp-port port-number;
      collector-vrf-name vrf-name;
    }
  }
}
```

Hierarchy Level

```
[edit services analytics agent service-agents agent-name]
```

Release Information

Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.
kafka and **file** options added in Junos OS Release 18.4R1 on MX Series and PTX Series routers.

Description

Configure parameters for the Network Telemetry Framework (NTF) agent output plug-in.

NOTE: When you modify the output plug-in configuration of a service agent instance, the associated service agent daemon is restarted.

Options

file parameters—Configure parameters for sending data in a log file to a data collector.

path ***pathname***—Path for the log file to which to save the data. For example, **path /tmp/example_file.log**

kafka parameters—Configure parameters for sending data to a Kafka data collector.

server *ip-address*—IP address of the Kafka server.

topic *filename*—Kafka topic name. The naming convention of the topic is *server-name*.jti.*encoding-type*. The encoding type options are **avro**, **json**, or **msgpack**.

encoding *encoding-type*—Encoding type. Options are **avro**, **json**, or **msgpack**.

output-ipfix parameters—Configure parameters for the IPFIX mediation service agent to send the IPFIX records that have been consolidated on the router to the IPFIX collector.

You must configure the IP address of the upstream IPFIX collector. When you optionally configure at least one of the collector certificate options (**collector-ca-certificate**, **collector-certificate**, and **collector-certificate-key**), the IPFIX mediator attempts to use TLS to connect with the collector. Otherwise, the mediator uses a TCP connection.

NOTE: Any change you make to an existing output-ipfix output plug-in configuration restarts the IPFIX service agent daemon to apply the changes.

collector-address *ip-address*—IP address of the upstream IPFIX collector.

collector-ca-certificate *file-path*—(Optional) Path for the certificate, provided by a trusted certificate authority (CA), that is used to sign the peer certificate at the peer (IPFIX collector) level. The certificate is expected to be in .pem container format.

collector-certificate *file-path*—(Optional) Path for the client certificate that the server (IPFIX collector) uses to authenticate the client and enable mutual authentication. The fully-qualified domain name (FQDN) of both the client and the server are stored in the certificate's Subject Alternative Name field when the client and server certificates are generated. The certificate is expected to be in .pem container format.

collector-certificate-key *file-path*—(Optional) Private key file that is loaded to decrypt the encrypted message sent from the peer.

collector-connection-retry-interval *seconds*—(Optional) Interval in seconds at which the output plug-in retries connecting to the IPFIX collector.

Range: 1 through 25

Default: 20

collector-tcp-port *port-number*—(Optional) Number of the TCP port used to connect to the IPFIX collector.

Default: 4740

collector-vrf-name *vrf-name*—(Optional) Name of the VRF (routing instance) in which IPFIX packets are routed.

Default: default

Required Privilege Level system

Related Documentation

- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- Configuring NTF Agent on page 212
- *IPFIX Mediation on the BNG*
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

service-agents (Analytics)

Syntax

```
service-agents {
  agent-name {
    inputs {
      analytics {
        parameters {
          generate-tags value;
          sample-frequency value;
          sensors file-path;
        }
      }
      input-ipfix {
        parameters {
          maximum-connections number;
          tcp-port port-number;
          vrf-name name;
        }
      }
      input-jti-ipfix {
        parameters {
          record-group group-name {
            record ipfix-record-name;
            reporting-interval seconds;
          }
        }
      }
    }
    outputs {
      file {
        parameters {
          path file-path;
        }
      }
      kafka {
        parameters {
          server ip-address;
          topic topic-name;
          encoding encoding-type;
        }
      }
      output-ipfix {
        parameters {
          collector-address ip-address;
          collector-ca-certificate file-path;
          collector-certificate file-path;
          collector-certificate-key file-path;
          collector-connection-retry-interval seconds;
          collector-tcp-port port-number;
          collector-vrf-name vrf-name;
        }
      }
    }
```

```
  }
}
```

Hierarchy Level

[edit services analytics agent]

Release Information

Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.

Description

Configure a network analytics service agent that uses input and output plug-ins to collect, transform, and forward network telemetry data.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

system

Related Documentation

- *Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data*
- *Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator*
- Configuring NTF Agent on page 212
- *IPFIX Mediation on the BNG*
- *Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector*

show services analytics agent

Syntax

```
show services analytics agent
<brief | detail>
```

Release Information

Command introduced in Junos OS Release 18.3R1 on MX Series routers.
Command introduced in Junos OS Release 18.4R1 on PTX Series routers.

Description

Display information about running instances of Network Telemetry Framework (NTF) agent.

Options

none—(Same as brief) Display summary information about analytics agents.

brief | detail—(Optional) Display information about analytics agents for the specified level of output.

Required Privilege Level

view

Related Documentation

- *IPFIX Mediation on the BNG*
- Configuring NTF Agent on page 212

List of Sample Output

show services analytics agent on page 296
show services analytics agent (Brief) on page 296
show services analytics agent (Detail) on page 296

Output Fields

Table 12 on page 295 lists the output fields for the **show services analytics agent** command. Output fields are listed in the approximate order in which they appear.

Table 12: show services analytics agent Output Fields

Field Name	Field Description	Level of Output
Agent ID	Name of the agent.	**brief** none
Output Plugins	Number of output plug-ins configured for the agent.	**brief** none
Input Plugins	Number of input plug-ins configured for the agent.	**brief** none
Process ID	Number that uniquely identifies the active process for the service agent at the **brief** and none levels. At the **detail** level, the process ID is displayed for the analytics agent (the parent NTF agent) and for the active service agents.	All levels
Analytics agent	Information about the parent NTF agent.	**detail**
Configuration File	Path where the NTF agent configuration file is located.	**detail**

Table 12: show services analytics agent Output Fields (continued)

Field Name	Field Description	Level of Output
Log File	Path where logs are stored for the NTF agent.	**detail**
Service Agent Count	Number of active service agents.	**detail**
Analytics Service agent(s)	Information about the active service agents.	**detail**
Agent Name	Name of the service agent.	**detail**
Input Plugin/s	Name of all input plug-ins configured for the service agent.	**detail**
Output Plugin/s	Name of all output plug-ins configured for the service agent.	**detail**

Sample Output

show services analytics agent

```
user@host> show services analytics agent

Agent ID        Output Plugins    Input Plugins    Process ID
ipfix           1                 2                8368
```

show services analytics agent (Brief)

```
user@host> show services analytics agent brief
Agent ID        Output Plugins    Input Plugins    Process ID
ipfix           1                 2                8368
```

show services analytics agent (Detail)

```
user@host> show services analytics agent detail

Analytics agent:
Process ID          : 6246
Configuration File  : /var/etc/ntf-agent.conf
Log File            : /var/log/ntf-agent.log
Service Agent Count : 1
Analytics service agent(s):
  Agent Name        : ipfix
  Input Plugin/s    : input-ipfix
  Output Plugin/s   : output-ipfix
  Process ID        : 8368
```

traceoptions (Analytics Agent)

Syntax

```
traceoptions {
  file filename;
  flag (debug | error | info | trace);
}
```

Hierarchy Level

[edit services analytics agent]

Release Information

Statement introduced in Junos OS Release 18.3R1 on MX Series routers.
Statement introduced in Junos OS Release 18.4R1 on PTX Series routers.

Description

Configure tracing operations for Network Telemetry Framework (NTF) agent. You can specify the name of the file where the NTF agent log messages are stored. You can also specify a severity level for messages to be logged. The severity level that you configure depends on the issue that you are trying to resolve. In some cases you might be interested in seeing all messages relevant to the logged event, so you specify **trace**. As levels become more restrictive, fewer messages are logged.

NOTE: Although the syntax uses the keyword **flag**, its function in this statement corresponds to the **level** keyword used for other **traceoptions** statements.

Options

file *filename*—Name of the file to receive the output of the tracing operation. The file is stored in the **/var/log/** directory of your device.

Default: ntf-agent

flag (debug | error | info | trace)—Specify the severity level for messages to be logged. The order of severity, from most to least severe is as follows:

error > info > debug > trace

- **debug**—Match debug messages.
- **error**—Match error messages. This is the most restrictive level.
- **info**—Match informational messages.
- **trace**—Match all messages.

Default: error

Required Privilege Level

system

Related Documentation

- *IPFIX Mediation on the BNG*
- Configuring NTF Agent on page 212

CHAPTER 11

J-Insight Device Monitor Configuration Statements and Operational Commands

clear chassis fpc errors

Syntax clear chassis fpc errors fpc-slot *fpc-slot* (all | error-id *error-id*)

Release Information Command introduced in Junos OS Release 18.2R1.

Description Clear the chassis FPC errors. You can choose to clear a particular error or all errors on the FPC.

Options **fpc-slot *fpc-slot***—The slot number of the FPC in which you want to run this command.

all—Clear all the errors on the FPC.

error-id *error-id*—Clear a particular error identified by an error-id. An *error-id*, a unique error identifier, is represented as a Uniform Resource Identifier (URI). For example, **"/cpu/0/memory/0/memory-uncorrected-error"** is an error-id that indicates an uncorrectable error under CPU memory module instance **0**.

Required Privilege Level clear

Related Documentation
- fpc error on page 307

List of Sample Output clear chassis fpc errors on page 300

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

Sample Output

clear chassis fpc errors

```
user@host> clear chassis fpc errors fpc-slot 1 all

Clearing error(s) on fpc 1, option all
```

clear system errors

Syntax

```
clear system errors fpc fpc-slot fpc-slot
<all>
<error-id error-id-uri>
```

Release Information Command introduced in Junos OS Release 18.2R1.

Description Clear system errors associated with J-Insight fault monitoring.

Options

all—(Optional) Clear all systems errors.

error-id *error-id-uri*—(Optional) Clear system errors for a specified error ID URI.

fpc-slot *fpc-slot*—Clear system errors for a specified FPC.

Required Privilege Level clear

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- show system errors active on page 332
- show system errors count on page 337
- show system errors fru on page 341

Output Fields This command produces no output.

clear trace

Syntax **clear trace**

Release Information Command introduced in Junos OS Evolved Release 18.3R1.

Description Clear traces on the system. Trace data from all nodes is collected on the master Routing Engine in **/var/log/traces**. All applications are traced at the info level for informational messages.

Options This command has no options.

Required Privilege Level view

Related Documentation

- show trace on page 378

Sample Output

clear trace

```
user@host> clear trace
```

delete services jinsightd subscribe health-monitor

Syntax **delete services jinsightd subscribe health-monitor**

Release Information Command introduced in Junos OS Release 18.2R1.

Description Disables the J-Insight health monitor. Starting in Junos OS Release 18.2R1, J-Insight provides health monitoring capabilities for FPC FRUs on the MX series routers. The health monitor is disabled by default.

Options This command has no options.

Required Privilege Level system

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- set services jinsightd subscribe health-monitor on page 310

error

Syntax

```
error {
  (fatal | major | minor) {
    threshold threshold value;
    action (alarm | disable-pfe | offline-pic | log | get-state |offline | reset);
  }
  scope error-scope {
    category category {
      (fatal | major | minor) {
        threshold threshold value;
        action (alarm | disable-pfe | log | get-state |offline | reset);
      }
    }
  }
}
```

Hierarchy Level [edit chassis]

Release Information Statement introduced in Junos OS Release 13.3 on MX Series routers.

Description Configure the threshold at which FPC errors will take the action you configure to be performed by the device. Starting from Junos OS Release 18.1R3, you can configure error thresholds and actions at the error scope and error category levels on MX Series routers.

Some Juniper devices include an internal framework for detecting and correcting FPC errors that can have the potential to affect services. You can classify FPC errors according to severity, set an automatic recovery action for each severity, and set a threshold (i.e., the number of times the error must occur before the action is triggered).

Options You can configure the threshold for the following severity levels:

NOTE: You cannot configure the severity level of an error. However you can modify the severity of an error by using the error ID. See *error-id*.

- **fatal**—Fatal error on the FPC. An error that results in blockage of considerable amount of traffic across modules is a fatal error.
- **major**—Major error on the FPC. An error that results in continuing loss of packet traffic but does not affect other modules is a major error.
- **minor**—Minor error on the FPC. An error that results in the loss of a single packet but is fully recoverable is a minor error.
- **threshold *threshold-value***—Configure the threshold value at which to take action. If the severity level of the error is fatal, the action is carried out only once when the total number of errors crosses the threshold value. If the severity level of the error is major,

the action is carried out once after the occurrence crosses the threshold. If the severity level is minor, the action is carried out as many times as the value specified by the threshold. For example, when the severity level is minor, and you have configured the threshold value as 10, the action is carried out after the tenth occurrence.

NOTE: You can set the threshold value to 0 for errors with severity level as minor. This implies that no action is taken for that error. You cannot set the threshold value to 0 for errors with severity level as major or fatal.

Default: The error count for fatal and major actions is 1. The default error count for minor actions is 10.

Range: 0—429,496,729

The available detection and recovery actions are as follows:

- **alarm**—Raise an alarm.
- **disable-pfe**—Disable the PFE interfaces on the FPC.
- **get-state**—Get the current state of the FPC.
- **log**—Generate a log for the event.
- **offline**—Take the FPC offline.
- **offline-pic**—Take the PIC (installed in the FPC) offline.
- **reset**—Reset the FPC.

NOTE: Starting in Junos OS Evolved Release 19.1R1, the **offline** and **disable-pfe** actions are not available for errors with **minor** severity (under the hierarchy **edit chassis error minor action**).

The available detection and recovery actions are as follows for devices running Junos OS Evolved:

- **alarm**—Raise an alarm.
- **fault**—System goes to fault state but stays up (diagnostics can be run on it).
- **get-state**—Get the current state of the FPC.
- **log**—Generate a log for the event.

NOTE: Starting in Junos OS Release 17.2R1, if you configure the **disable-pfe**, **offline**, **offline-pic** or **reset** action on an MX Series or PTX Series router, the **get-state** action is additionally configured on the router. This means, for example, if you configure the **disable-pfe** action on the router, the router gets both **disable-pfe** and **get-state** actions configured.

- **scope** ***error-scope***—Group the errors of a particular severity into different scopes. Errors belonging to each error scope is further grouped into categories, before thresholds and actions are defined at the group level. The following scopes are available: **board** and **pfe**.
- **category** ***category***—Categorize errors into various subgroups under the scope level. An error category helps you group similar errors belonging to a particular scope and define actions for them at once. This feature eliminates the need for configurations against individual error-ids. Some of the error-categories are **functional**, **io** (input/output errors), **storage** (for example, errors related to HDD, SSD, and flash), **memory** (for example, errors related to static RAM), **processing** (for example, CPU-related errors), and **switch**.

Required Privilege Level

interface—To view this statement in the configuration.

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

Related Documentation

- *Fabric Resiliency and Degradation*
- *Configuring FPC Error Levels and Actions*
- fpc error on page 307
- *show chassis fabric errors*
- *show chassis fpc errors*

fpc error

Syntax

```
fpc slot number {
  error {
    (fatal | major | minor) {
      threshold threshold value;
      action (alarm | disable-pfe | offline-pic | log | get-state |offline | reset);
    }
    scope error-scope {
      category category {
        (fatal | major | minor) {
          threshold threshold value;
          action (alarm | disable-pfe | log | get-state |offline | reset);
        }
      }
    }
  }
}
```

Hierarchy Level

```
[edit chassis]
```

Release Information

Statement introduced in Junos OS Release 13.3 on MX Series, PTX Series, and T Series routers.
Statement introduced in Junos OS Release 14.2 on M320 routers.

Description

Configure the threshold at which FPC errors will take the action you configure to be performed by the device. Starting from Junos OS Release 18.1R3, you can configure error thresholds and actions at the error scope and error category levels on MX Series routers.

Some Juniper devices include an internal framework for detecting and correcting FPC errors that can have the potential to affect services. For each FPC on the device, you can classify errors according to severity, set an automatic recovery action for each severity, and set a threshold (i.e., the number of times the error must occur before the action is triggered).

Options

You can configure the threshold for the following severity levels:

- **fatal**—Fatal error on the FPC. An error that results in blockage of considerable amount of traffic across modules is a fatal error.
- **major**—Major error on the FPC. An error that results in continuing loss of packet traffic but does not affect other modules is a major error.
- **minor**—Minor error on the FPC. An error that results in the loss of a single packet but is fully recoverable is a minor error.

NOTE: You cannot configure the severity level of an error.

- **threshold** ***threshold-value***—Configure the threshold value at which to take action. If the severity level of the error is fatal or major, the action is carried out when the total number of errors reaches the threshold value. After the threshold value is crossed, for every occurrence of the error, an action is carried out. If the severity level is minor, the action is carried out only once after the total number of errors crosses the threshold value.

  NOTE: You can set the threshold value to 0 for errors with severity level as minor. This implies that no action is taken for that error. You cannot set the threshold value to 0 for errors with severity level as major or fatal.

  Default: The error count for fatal and major actions is 1. The default error count for minor actions is 10.

  Range: 0—429,496,729

  Range(Junos OS Evolved): 0—1024

The available detection and recovery actions are as follows:

- **alarm**—Raise an alarm.
- **disable-pfe**—Disable the PFE interfaces on the FPC.

  NOTE: For PTX Series routers, when an alarm occurs and a disable-pfe action is the result, to clear the alarm you must place the FPC offline and then back online.

- **get-state**—Get the current state of the FPC.
- **log**—Generate a log for the event.
- **offline**—Take the FPC offline.
- **offline-pic**—Take the PIC (installed in the FPC) offline.

  NOTE: This option is supported only on T Series Routers.

- **reset**—Reset the FPC.

NOTE: Starting in Junos OS Release 17.2R1, if you configure the **disable-pfe**, **offline**, or **reset** action on an MX Series or PTX Series router, the **get-state** action is additionally configured on the router. This means, for example, if you configure the **disable-pfe** action on the router, the router gets both **disable-pfe** and **get-state** actions configured.

- **scope** ***error-scope***—Group the errors of a particular severity into different scopes. Errors belonging to each error scope is further grouped into categories, before thresholds and actions are defined at the category level. The following scopes are available: **board** and **pfe**.
- **category** ***category***—Categorize errors into various subgroups under the scope level. An error category helps you group similar errors belonging to a particular scope and define actions for them at once. This feature eliminates the need for configurations against individual error-ids. Some of the error-categories are **functional**, **io** (input/output errors), **storage** (for example, errors related to HDD, SSD, and flash), **memory** (for example, errors related to static RAM), **processing** (for example, CPU-related errors), and **switch**.
- ***error-id***—Use the error ID to disable an error or modify the error severity associated with that error. An *error-id*, which is a unique error identifier, is represented as a Uniform Resource Identifier (URI). For example, **/cpu/0/memory/0/memory-uncorrected-error** is an error ID that indicates an uncorrectable error under CPU memory module instance **0**.

Required Privilege Level

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

Related Documentation

- *Fabric Resiliency and Degradation*
- *Configuring FPC Error Levels and Actions*
- *show chassis fabric errors*
- *show chassis fpc errors*
- error on page 304

set services jinsightd subscribe health-monitor

Syntax **set services jinsightd subscribe health-monitor**

Release Information Command introduced in Junos OS Release 18.2R1.

Description Enables the J-Insight health monitor. Starting in Junos OS Release 18.2R1, J-Insight provides health monitoring capabilities for FPC FRUs on the MX series routers. The health monitor is disabled by default.

Options This command has no options.

Required Privilege Level system

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- delete services jinsightd subscribe health-monitor on page 303

set services jinsightd traceoptions

Syntax

```
set services jinsightd traceoptions flag
<all>
<core>
<database>
<rule-engine>
<timer>
```

Release Information Command introduced in Junos OS Release 18.2R1.

Description Define tracing operations that track J-Insight functionality. To specify more than one tracing operation, include multiple **flag** statements.

Options

all—All tracing operations.

core—J-Insight core events.

database—Database events.

rule-engine—Rule engine events.

timer—Timer events.

Required Privilege Level system

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222

show chassis alarms

List of Syntax

Syntax

```
show chassis alarms
```

Syntax (TX Matrix Routers)

```
show chassis alarms
<lcc number | scc>
```

Syntax (TX Matrix Plus Routers)

```
show chassis alarms
<lcc number | sfc number>
```

Syntax (MX Series Routers)

```
show chassis alarms
<all-members>
<local>
<member member-id>
```

Syntax (MX104, MX2010, MX2020, and MX2008 Universal Routing Platforms)

```
show chassis alarms
<satellite [slot-id slot-id]>
```

Syntax (MX10003, MX204, and MX10008)

```
show chassis alarms
```

Syntax (QFX Series)

```
show chassis alarms
<interconnect-device name>
<node-device name>
```

Syntax (OCX Series)

```
show chassis alarms
```

Syntax (PTX Series Packet Transport Routers)

```
show chassis alarms
```

Syntax (ACX Series Universal Metro Routers)

```
show chassis alarms
```

Syntax (EX9251, EX9253 Switches)

```
show chassis alarms
```

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
sfc option introduced in Junos OS Release 9.6 for the TX Matrix Plus router.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Command introduced in Junos OS Release 12.1 for the PTX Series Packet Transport Routers.
Command introduced in Junos OS Release 12.2 for the ACX Series Universal Metro Routers.
Command introduced in Junos OS Release 12.3 for MX 2010 and MX2020 Universal Routing Platforms.
Command introduced in Junos OS Release 13.2 for MX104 Universal Routing Platforms.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
satellite option introduced in Junos OS Release 14.2R3 for Junos Fusion.
Command introduced in Junos OS Release 17.2 for MX2008 Universal Routing Platforms.
Command introduced in Junos OS Release 17.2 for PTX10008 Routers.
Command introduced in Junos OS Release 17.3 for MX150 Router Appliance.
Command introduced in Junos OS Release 17.3 for MX10003 Universal Routing Platforms.
Command introduced in Junos OS Release 17.4 for MX204 Universal Routing Platforms.
Command introduced in Junos OS Release 18.1R1 for EX9251 Switches.
Command introduced in Junos OS Release 18.2 for EX9253 Switches.
Command introduced in Junos OS Release 18.2R1 for MX10008 Universal Routing Platforms.

Description

Display information about the conditions that have been configured to trigger alarms.

Options

none—Display information about the conditions that have been configured to trigger alarms.

all-members—(MX Series routers only) (Optional) Display information about alarm conditions for all the member routers of the Virtual Chassis configuration.

interconnect-device ***name***—(QFabric systems only) (Optional) Display information about alarm conditions for the Interconnect device.

lcc ***number***—(TX Matrix router and TX Matrix Plus router only) (Optional) Line-card chassis number.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

local—(MX Series routers only) (Optional) Display information about alarm conditions for the local Virtual Chassis member.

member *member-id*—(MX Series routers only) (Optional) Display information about alarm conditions for the specified member of the Virtual Chassis configuration. Replace *member-id* variable with a value of 0 or 1.

node-device *name*—(QFabric systems only) (Optional) Display information about alarm conditions for the Node device.

satellite [slot-id *slot-id*]—(Junos Fusion only)(Optional) Display information about alarm conditions for the specified satellite device in a Junos Fusion, or for all satellite devices in the Junos Fusion if no satellite devices are specified.

scc—(TX Matrix router only) (Optional) Show information about the TX Matrix router (switch-card chassis).

sfc *number*—(TX Matrix Plus router only) (Optional) Show information about the respective TX Matrix Plus router, which is the switch-fabric chassis. Replace *number* variable with **0**.

Additional Information

Chassis alarms are preset. You cannot modify them.

You cannot clear the alarms for chassis components. Instead, you must remedy the cause of the alarm. When a chassis alarm LED is lit, it indicates that you are running the router or switch in a manner that we do not recommend.

On routers, you can manually silence external devices connected to the alarm relay contacts by pressing the alarm cutoff button, located on the craft interface. Silencing the device does not remove the alarm messages from the display (if present on the router) or extinguish the alarm LEDs. In addition, new alarms that occur after you silence an external device reactivate the external device.

NOTE: MX10003 routers do not support craft interface.

In Junos OS release 11.1 and later, alarms for fans also show the slot number of the fans in the CLI output.

In Junos OS Release 11.2 and later, the command output on EX8200 switches shows the detailed location (**Plane/FPC/PFE**) for link errors in the chassis.

In Junos OS Release 10.2 and later, an alarm is shown on T Series routers for a standby SONET Clock Generator (SCG) that is offline or absent.

You may often see the following error messages, in which only the error code is shown and no other information is provided:

```
Apr 12 08:04:10 send: red alarm set, device FPC 6, reason FPC 6 Major Errors
 - Error code: 257
Apr 12 08:04:19 send: red alarm set, device FPC 1, reason FPC 1 Major Errors
 - Error code: 559
```

To understand what CM_ALARM error codes mean, you need to first identify the structure of the CM Alarm codes. A CM_ALARM code has the following structure:

Bits:	Error type:
1-31	Major (1)
0	Minor (0)

According to the table above, the LSB (bit 0) identifies the **Error Type** (major alarm, if the bit is set and minor alarm if the bit is unset). The rest of the bits (1 - 31) identify the actual error code.

Take an example of the following error code, which was logged on a T1600:

```
Apr 12 08:04:10 send: red alarm set, device FPC 1, reason FPC 1 Major Errors
 - Error code: 559
```

First, you have to convert 559 to binary; that is **1000101111**. The LSB in this case is **1**, which means that this is a major alarm. After removing the LSB, you are left with **100010111**, which is equal to 279 in decimal. This is the actual error code, its meaning can be found from the following list:

Chip Type: L Chip	**Code**
CMALARM_LCHIP_LOUT_DESRD_PARITY_ERR	1
CMALARM_LCHIP_LOUT_DESRD_UNINIT_ERR	2
CMALARM_LCHIP_LOUT_DESRD_ILLEGALLINK_ERR	3
CMALARM_LCHIP_LOUT_DESRD_ILLEGALSIZE_ERR	4
CMALARM_LCHIP_LOUT_HDRF_TOERR_ERR	5

CMALARM_LCHIP_LOUT_HDRF_PARITY_ERR	6
CMALARM_LCHIP_LOUT_HDRF_UCERR_ERR	7
CMALARM_LCHIP_LOUT_NLIF_CRCDROP_ERR	8
CMALARM_LCHIP_LOUT_NLIF_CRCERR_ERR	9
CMALARM_LCHIP_UCODE_TIMEOUT_ERR	10
CMALARM_LCHIP_LIN_SRCTL_ACCT_DROP_ERR	11
CMALARM_LCHIP_LIN_SRCTL_ACCT_ADDR_SIZE_ERR	12
CMALARM_LCHIP_SRAM_PARITY_ERR	13
CMALARM_LCHIP_UCODE_OVFLW_ERR	14
CMALARM_LCHIP_LOUT_HDRF_MTU_ERR	15

Chip Type: M Chip	Code
CMALARM_MCHIP_ECC_UNCORRECT_ERR	128

Chip Type: N Chip	Code
CMALARM_NCHIP_RDDMA_JBUS_TIMEOUT_ERR	256
CMALARM_NCHIP_RDDMA_FIFO_OVFLW_ERR	257
CMALARM_NCHIP_RDDMA_FIFO_UNFLW_ERR	258
CMALARM_NCHIP_RDDMA_SIZE_ERR	259
CMALARM_NCHIP_RDDMA_JBUS_CRC_ERR	260
CMALARM_NCHIP_WRDMA_PKTR_ERR	261
CMALARM_NCHIP_WRDMA_PKT_CRC_ERR	262
CMALARM_NCHIP_WRDMA_JBUS_TIMEOUT_ERR	263
CMALARM_NCHIP_WRDMA_FIFO_OVFLW_ERR	264
CMALARM_NCHIP_WRDMA_FIFO_UNFLW_ERR	265
CMALARM_NCHIP_WRDMA_PKT_LEN_ERR	266
CMALARM_NCHIP_WRDMA_JBUS_CRC_ERR	267

CMALARM_NCHIP_PKTR_DMA_AGE_ERR	268
CMALARM_NCHIP_PKTR_ICELLSIG_ERR	269
CMALARM_NCHIP_PKTR_FTTL_ERR	270
CMALARM_NCHIP_RODR_OFFSET_OVFLW_ERR	271
CMALARM_NCHIP_PKTR_TMO_CELL_ERR	272
CMALARM_NCHIP_PKTR_TMO_OUTRANGE_ERR	273
CMALARM_NCHIP_PKTR_MD_REQUEST_Q_OVFLW_ERR	274
CMALARM_NCHIP_PKTR_DMA_BUFFER_OVFLW_ERR	275
CMALARM_NCHIP_PKTR_GRT_OVFLW_ERR	276
CMALARM_NCHIP_FRQ_ERR	277
CMALARM_NCHIP_RODR_IN_Q_OVFLW_ERR	278
CMALARM_NCHIP_DBUF_CRC_ERR	279

Chip Type: R Chip	Code
CMALARM_RCHIP_SRAM_PARITY_ERR	512

Chip Type: R Chip	Code
CMALARM_ICHIP_WO_DESRD_ID_ERR	601
CMALARM_ICHIP_WO_DESRD_DATA_ERR	602
CMALARM_ICHIP_WO_DESRD_OFLOW_ERR	603
CMALARM_ICHIP_WO_HDRF_UCERR_ERR	604
CMALARM_ICHIP_WO_HDRF_MTUERR_ERR	605
CMALARM_ICHIP_WO_HDRF_PARITY_ERR	606
CMALARM_ICHIP_WO_HDRF_TOERR_ERR	607
CMALARM_ICHIP_WO_IP_CRC_ERR	608
CMALARM_ICHIP_WO_IP_INTER_ERR	609
CMALARM_ICHIP_WI_WAN_TIMEOUT_ERR	625

CMALARM_ICHIP_WI_FAB_TIMEOUT_ERR	626
CMALARM_ICHIP_RLDRAM_BIST_ERR	630
CMALARM_ICHIP_SDRAM_BIST_ERR	631
CMALARM_ICHIP_RLDRAM_PARITY_ERR	632
CMALARM_ICHIP_SDRAM_UNCORRECT_ERR	633
CMALARM_ICHIP_SDRAM_CORRECT_ERR	634
CMALARM_ICHIP_FUSE_DONE_ERR	635

According to the table above, the **279** error code corresponds to **CMALARM_NCHIP_DBUF_CRC_ERR**; this means that new CRC errors were seen on the NCHIP of this particular FPC, which is FPC as per the logs.

If you do not want to convert decimal to binary and vice versa, you may use the following shortcut:

For major alarms, the **Actual Error Code = (Error Code - 1)/2**, where **Error Code** is the code that you get in the log message. For example, if you get the following log:

```
    Apr 12 08:04:10 send: red alarm set, device FPC 6, reason FPC 6 Major
Errors - Error code: 257
```

Actual Error Code = (257-1)/2 = 128. Similarly, for minor alarms, Actual Error Code = (Error Code)/2

NOTE: Starting in Junos OS Release 18.2R1, on MX Series routers, the **show chassis alarms** output does not display error codes for PFE-related errors. You can use the following commands to view more details of the errors that caused the alarms:

- **show chassis errors active**
- **show chassis errors active detail**

Required Privilege Level view

Related Documentation

- *Configuring an RMON Alarm Entry and Its Attributes*
- *Chassis Conditions That Trigger Alarms*

List of Sample Output

Output Fields Table 13 on page 320 lists the output fields for the **show chassis alarms** command. Output fields are listed in the approximate order in which they appear.

Table 13: show chassis alarms Output Fields

Field Name	Field Description
Alarm time	Date and time the alarm was first recorded.
Class	Severity class for this alarm: **Minor** or **Major**.
Description	Information about the alarm.

Sample Output

show chassis alarms (Alarms Active)

```
user@host> show chassis alarms
3 alarms are currently active
Alarm time               Class  Description
2000-02-07 10:12:22 UTC Major fxp0: ethernet link down
2000-02-07 10:11:54 UTC Minor YELLOW ALARM - PEM 1 Removed
2000-02-07 10:11:03 UTC Minor YELLOW ALARM - Lower Fan Tray Removed
```

show chassis alarms (No Alarms Active)

```
user@host> show chassis alarms
No alarms are currently active
```

show chassis alarms (Fan Tray)

```
user@host> show chassis alarms
4 alarms currently active
Alarm time               Class  Description
2010-11-11 20:27:38 UTC  Major  Side Fan Tray 7 Failure
2010-11-11 20:27:13 UTC  Minor  Side Fan Tray 7 Overspeed
2010-11-11 20:27:13 UTC  Major  Side Fan Tray 5 Failure
2010-11-11 20:27:13 UTC  Major  Side Fan Tray 0 Failure
```

show chassis alarms (MX150)

```
user@host > show chassis alarms
1 alarms currently active
Alarm time               Class  Description
2016-06-04 01:49:43 PDT  Major  Fan Tray 1 Fan 0 failed
```

show chassis alarms (MX104 Router)

```
user@host >show chassis alarms
```

```
1 alarms currently active
Alarm time               Class  Description
2013-06-05 14:43:31 IST  Minor  Backup RE Active
```

show chassis alarms (MX2010 Router)

```
user@host> show chassis alarms
```

```
7 alarms currently active
Alarm time               Class  Description
2012-08-07 00:46:06 PDT  Major  Fan Tray 2 Failure
2012-08-06 18:24:36 PDT  Minor  Redundant feed missing for PSM 6
2012-08-06 07:41:04 PDT  Minor  Redundant feed missing for PSM 8
2012-08-04 02:42:06 PDT  Minor  Redundant feed missing for PSM 5
2012-08-03 21:14:24 PDT  Minor  Loss of communication with Backup RE
2012-08-03 12:26:03 PDT  Minor  Redundant feed missing for PSM 4
2012-08-03 10:40:18 PDT  Minor  Redundant feed missing for PSM 7
```

show chassis alarms (MX2020 Router)

```
user@host> show chassis alarms
```

```
1 alarms currently active
Alarm time Class Description
2012-10-03 12:14:59 PDT Minor Plane 0 not online
```

show chassis alarms (MX10003 Router)

```
user@host> show chassis alarms
```

```

9 alarms currently active
Alarm time               Class  Description
2017-07-13 21:50:31 PDT  Major  FPC 1 Temperature Hot
2017-07-13 21:50:04 PDT  Minor  FPC 1 PIC 1 Invalid port profile configuration
2017-07-13 21:49:13 PDT  Minor  FPC 1 PIC 0 Invalid port profile configuration
2017-07-13 21:48:54 PDT  Major  FPC 0 Temperature Hot
2017-07-13 21:43:57 PDT  Minor  PEM 5 Not Present
2017-07-13 21:43:57 PDT  Minor  PEM 4 Not Present
2017-07-13 21:43:54 PDT  Minor  CB 1 Voltage Sensor ADS7830_0x4B Sensor Failed
2017-07-13 21:43:54 PDT  Minor  CB 0 Voltage Sensor ADS7830_0x4B Sensor Failed
2017-07-13 21:43:31 PDT  Minor  Loss of communication with Backup RE
```

show chassis alarms (MX204 Router)

```
user@host> show chassis alarms
```

```

1 alarms currently active
Alarm time               Class  Description
2017-11-05 22:13:03 PST  Major  PEM 0 Not Present
```

show chassis alarms (MX2008 Router)

```
user@host>show chassis alarms
```

```
No alarms currently active
```

show chassis alarms (MX960, MX480, and MX240 Routers showing Major CB Failure)

A major CB 0 failure alarm occurs in the event of a bad CB (unknown or mismatched CBs do not trigger this alarm in Junos Release OS 12.3R9 and later). Following GRES or recovery, if the hardware issue persists, the traffic moves to the good CB and continues. If the alarm was triggered by something transient like a power zone budget on GRES, bringing the CB back online can clear the alarm. Otherwise, replace the bad CB. Note that fabric link speed is not impacted by an offline SCB. The alarm migh be raised on CB0, CB1, and CB2.

```
user@host> show chassis alarms

6 alarms currently active
Alarm time               Class  Description
2014-10-31 16:49:41 EDT  Major  PEM 3 Not OK
2014-10-31 16:49:41 EDT  Major  PEM 2 Not OK
2014-10-31 16:49:31 EDT  Major  CB 0 Failure
2014-10-31 16:49:31 EDT  Minor  CB 0 Fabric Chip 0 Not Online
2014-10-31 16:49:31 EDT  Minor  CB 0 Fabric Chip 1 Not Online
2014-10-31 16:49:31 EDT  Minor  Backup RE Active
```

show chassis alarms (PTX10008 Router)

```
user@host>show chassis alarms

12 alarms currently active
Alarm time               Class  Description
2017-05-09 01:38:55 PDT  Minor  Loss of communication with Backup RE
2017-05-05 06:49:57 PDT  Major  FPC 5 LCPU Temp Sensor Access Failed
2017-05-05 06:49:57 PDT  Major  FPC 5 PE2 Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 PE1 Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 PE0 Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 Exhaust-C Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 Exhaust-B Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 Exhaust-A Temp Sensor Hot
2017-05-05 06:49:57 PDT  Major  FPC 5 Intake-B Temp Sensor Access Failed
2017-05-05 06:49:57 PDT  Major  FPC 5 Intake-A Temp Sensor Access Failed
2017-05-05 06:49:57 PDT  Major  Fan Tray 0 Fan 5 running at lower speed
2017-05-05 06:49:57 PDT  Major  Fan Tray 0 Fan 4 running at lower speed
```

show chassis alarms (T4000 Router)

```
user@host> show chassis alarms

9 alarms currently active
Alarm time               Class  Description
2007-06-02 01:41:10 UTC  Minor  RE 0 Not Supported
2007-06-02 01:41:10 UTC  Minor  CB 0 Not Supported
2007-06-02 01:41:10 UTC  Minor  Mixed Master and Backup RE types
2007-05-30 19:37:33 UTC  Major  SPMB 1 not online
2007-05-30 19:37:29 UTC  Minor  Front Bottom Fan Tray Absent
2007-05-30 19:37:13 UTC  Major  PEM 1 Input Failure
2007-05-30 19:37:13 UTC  Major  PEM 0 Not OK
2007-05-30 19:37:03 UTC  Major  PEM 0 Improper for Platform
2007-05-30 19:37:03 UTC  Minor  Backup RE Active
```

show chassis alarms (Unreachable Destinations Present on a T Series Router)

```
user@host> show chassis alarms
10 alarms currently active
Alarm time               Class  Description
2011-08-30 18:43:53 PDT  Major  FPC 7 has unreachable destinations
2011-08-30 18:43:53 PDT  Major  FPC 5 has unreachable destinations
2011-08-30 18:43:52 PDT  Major  FPC 3 has unreachable destinations
2011-08-30 18:43:52 PDT  Major  FPC 2 has unreachable destinations
2011-08-30 18:43:52 PDT  Minor  SIB 0 Not Online
2011-08-30 18:43:33 PDT  Minor  SIB 4 Not Online
2011-08-30 18:43:28 PDT  Minor  SIB 3 Not Online
2011-08-30 18:43:05 PDT  Minor  SIB 2 Not Online
2011-08-30 18:43:28 PDT  Minor  SIB 1 Not Online
2011-08-30 18:43:05 PDT  Major  PEM 1 Not Ok
```

show chassis alarms (FPC Offline Due to Unreachable Destinations on a T Series Router)

```
user@host> show chassis alarms
10 alarms currently active
Alarm time               Class  Description
2011-08-30 18:43:53 PDT  Major  FPC 7 offline due to unreachable destinations
2011-08-30 18:43:53 PDT  Major  FPC 5 offline due to unreachable destinations
2011-08-30 18:43:52 PDT  Major  FPC 3 offline due to unreachable destinations
2011-08-30 18:43:52 PDT  Major  FPC 2 offline due to unreachable destinations
2011-08-30 18:43:52 PDT  Minor  SIB 0 Not Online
2011-08-30 18:43:33 PDT  Minor  SIB 4 Not Online
2011-08-30 18:43:28 PDT  Minor  SIB 3 Not Online
2011-08-30 18:43:05 PDT  Minor  SIB 2 Not Online
2011-08-30 18:43:28 PDT  Minor  SIB 1 Not Online
2011-08-30 18:43:05 PDT  Major  PEM 1 Not Ok
```

show chassis alarms (SCG Absent on a T Series Router)

```
user@host> show chassis alarms
4 alarms currently active
Alarm time               Class  Description
2011-01-23 21:42:46 PST  Major  SCG 0 NO EXT CLK MEAS-BKUP SCG ABS
```

show chassis alarms (Alarms Active on a TX Matrix Router)

```
user@host> show chassis alarms
scc-re0:
--------------------------------------------------------------------------
8 alarms currently active
Alarm time               Class  Description
2004-08-05 18:43:53 PDT  Minor  LCC 0 Minor Errors
2004-08-05 18:43:53 PDT  Minor  SIB 3 Not Online
2004-08-05 18:43:52 PDT  Major  SIB 2 Absent
2004-08-05 18:43:52 PDT  Major  SIB 1 Absent
2004-08-05 18:43:52 PDT  Major  SIB 0 Absent
2004-08-05 18:43:33 PDT  Major  LCC 2 Major Errors
2004-08-05 18:43:28 PDT  Major  LCC 0 Major Errors
2004-08-05 18:43:05 PDT  Minor  LCC 2 Minor Errors
lcc0-re0:
--------------------------------------------------------------------------
```

```
5 alarms currently active
Alarm time               Class  Description
2004-08-05 18:43:53 PDT  Minor  SIB 3 Not Online
2004-08-05 18:43:49 PDT  Major  SIB 2 Absent
2004-08-05 18:43:49 PDT  Major  SIB 1 Absent
2004-08-05 18:43:49 PDT  Major  SIB 0 Absent
2004-08-05 18:43:28 PDT  Major  PEM 0 Not OK
lcc2-re0:
--------------------------------------------------------------------------
5 alarms currently active
Alarm time               Class  Description
2004-08-05 18:43:35 PDT  Minor  SIB 3 Not Online
2004-08-05 18:43:33 PDT  Major  SIB 2 Absent
2004-08-05 18:43:33 PDT  Major  SIB 1 Absent
2004-08-05 18:43:33 PDT  Major  SIB 0 Absent
2004-08-05 18:43:05 PDT  Minor  PEM 1 Absent
```

show chassis alarms (TX Matrix Plus router with 3D SIBs)

```
user@host> show chassis alarms
```

```
sfc0-re0:
--------------------------------------------------------------------------
Alarm time               Class  Description

2014-04-08 14:35:13 IST  Minor  FPM 0 SFC Config Size Changed
2014-04-08 14:32:58 IST  Major  Fan Tray Failure
2014-04-08 14:31:53 IST  Major  SIB F13 6 Fault
2014-04-08 14:31:43 IST  Major  SIB F13 11 Fault
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 14 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 8 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 3 Fbr Cbl
2014-04-08 14:31:08 IST  Major  SIB F13 12 CXP 15 fault
2014-04-08 14:31:08 IST  Minor  SIB F13 12 CXP 14 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 14
2014-04-08 14:31:08 IST  Major  SIB F13 12 CXP 10 fault
2014-04-08 14:31:08 IST  Minor  SIB F13 12 CXP 8 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 8
2014-04-08 14:31:08 IST  Major  SIB F13 12 CXP 7 fault
2014-04-08 14:31:08 IST  Major  SIB F13 12 CXP 4 fault
2014-04-08 14:31:08 IST  Minor  SIB F13 12 CXP 3 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 12 CXP 3
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 14 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 12 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 8 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 6 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 4 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 2 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 0 Fbr Cbl
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 14 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 14
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 12 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 12
2014-04-08 14:31:08 IST  Major  SIB F13 6 CXP 10 fault
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 8 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 8
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 6 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 6
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 4 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 4
```

```
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 2 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 2
2014-04-08 14:31:08 IST  Minor  SIB F13 6 CXP 0 LOL
2014-04-08 14:31:08 IST  Minor  Check SIB F13 6 CXP 0
2014-04-08 14:31:08 IST  Minor  SIB F13 12 CXP 14 XC HSL Link Error
2014-04-08 14:29:27 IST  Minor  LCC 0 Minor Errors
2014-04-08 14:28:37 IST  Major  LCC 0 Major Errors
2014-04-08 14:28:37 IST  Major  LCC 2 Major Errors
2014-04-08 14:28:37 IST  Minor  LCC 2 Minor Errors
2014-04-08 14:28:24 IST  Major  SIB F2S 4/6 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 4/4 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 4/2 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 4/0 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 3/6 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 3/4 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 3/2 Absent
2014-04-08 14:28:24 IST  Major  SIB F2S 3/0 Absent
2014-04-08 14:28:24 IST  Major  SIB F13 9 Absent
2014-04-08 14:28:24 IST  Major  SIB F13 8 Absent
2014-04-08 14:28:24 IST  Major  SIB F13 7 Absent
2014-04-08 14:28:24 IST  Major  SIB F13 4 Absent
2014-04-08 14:28:24 IST  Major  SIB F13 1 Absent
2014-04-08 14:28:22 IST  Major  PEM 0 Input Failure
2014-04-08 14:28:22 IST  Major  PEM 0 Not OK

lcc0-re0:
--------------------------------------------------------------------------
12 alarms currently active
Alarm time               Class  Description
2014-04-08 14:36:08 IST  Minor  CB 1  M/S Switch Changed
2014-04-08 14:36:08 IST  Minor  CB 1  CHASSIS ID Changed
2014-04-08 14:35:43 IST  Minor  CB 0  M/S Switch Changed
2014-04-08 14:35:43 IST  Minor  CB 0  CHASSIS ID Changed
2014-04-08 14:29:30 IST  Minor  SIB 4 Not Online
2014-04-08 14:29:30 IST  Minor  SIB 3 Not Online
2014-04-08 14:29:30 IST  Minor  SIB 2 Not Online
2014-04-08 14:29:24 IST  Major  Rear Fan Tray Failure
2014-04-08 14:29:24 IST  Major  Front Bottom Fan Tray Improper for Platform
2014-04-08 14:29:24 IST  Major  Front Top Fan Tray Improper for Platform
2014-04-08 14:28:37 IST  Major  SIB 4 Absent
2014-04-08 14:28:37 IST  Major  SIB 3 Absent

lcc2-re0:
--------------------------------------------------------------------------
12 alarms currently active
Alarm time               Class  Description
2014-04-08 14:36:02 IST  Minor  CB 1  M/S Switch Changed
2014-04-08 14:36:02 IST  Minor  CB 1  CHASSIS ID Changed
2014-04-08 14:35:42 IST  Minor  CB 0  M/S Switch Changed
2014-04-08 14:34:42 IST  Minor  CB 0  CHASSIS ID Changed
2014-04-08 14:29:29 IST  Minor  SIB 0 CXP 7 Unsupported Optics
2014-04-08 14:29:27 IST  Major  Front Bottom Fan Tray Improper for Platform
2014-04-08 14:29:27 IST  Major  Front Top Fan Tray Improper for Platform
2014-04-08 14:29:25 IST  Minor  SIB 4 Not Online
2014-04-08 14:29:25 IST  Minor  SIB 3 Not Online
2014-04-08 14:28:47 IST  Major  PEM 0 Not OK
2014-04-08 14:28:36 IST  Major  SIB 2 Absent
2014-04-08 14:28:36 IST  Minor  Host 0 Boot from alternate media

lcc6-re0:
```

```
------------------------------------------------------------------------
2 alarms currently active
Alarm time               Class  Description
2013-11-06 04:03:56 PST  Minor  SIB 1 CXP 0 XC HSL Link Error
2013-11-06 03:49:32 PST  Major  PEM 1 Not OK
```

show chassis alarms (Alarms on a T4000 Router After the enhanced-mode Statement is Enabled)

To enable improved virtual private LAN service (VPLS) MAC address learning on T4000 routers, you must include the **enhanced-mode** statement at the **[edit chassis network-services]** hierarchy level and reboot the router. When router reboots, only the T4000 Type 5 FPCs are required to be present on the router. If there are any other FPCs (apart from T4000 Type 5 FPCs) on the T4000 router, such FPCs become offline, and FPC misconfiguration alarms are generated. The **show chassis alarm** command output displays FPC misconfiguration (**FPC *fpc-slot* misconfig**) as the reason for the generation of the alarms.

```
user@host> show chassis alarms
2 alarms currently active
  Alarm time               Class  Description
  2011-10-22 10:10:47 PDT  Major  FPC 1 misconfig
  2011-10-22 10:10:46 PDT  Major  FPC 0 misconfig
```

show chassis alarms (Backup Routing Engine)

```
user@host> show chassis alarms
2 alarms are currently active
Alarm time               Class  Description
2005-04-07 10:12:22 PDT  Minor  Host 1 Boot from alternate media
2005-04-07 10:11:54 PDT  Major  Host 1 compact-flash missing in Boot List
```

show chassis alarms (EX Series Switch)

```
user@switch> show chassis alarms
 4 alarms currently active
Alarm time               Class  Description
2014-03-12 15:36:09 UTC  Minor  Require a Fan Tray upgrade
2014-03-12 15:00:02 UTC  Major  PEM 0 Input Failure
2014-03-12 15:00:02 UTC  Major  PEM 0 Not OK
2014-03-12 14:59:51 UTC  Minor  Host 1 Boot from alternate media
```

show chassis alarms (Alarms Active on the QFX Series and OCX Series Switches)

```
user@switch> show chassis alarms
1 alarms currently active
Alarm time               Class  Description
2012-03-05 2:10:24 UTC  Major  FPC 0 PEM 0 Airflow not matching Chassis Airflow
```

show chassis alarms node-device (Alarms Active on the QFabric System)

```
user@switch> show chassis alarms node-device Test
```

```
node-device ED3694
3 alarms currently active
Alarm time               Class  Description
2011-08-24 16:04:15 UTC  Major  Test:fte-0/1/2: Link down
2011-08-24 16:04:14 UTC  Major  Test:fte-0/1/0: Link down
2011-08-24 14:21:14 UTC  Major  Test PEM 0 is not supported/powered
```

show chassis alarms (Alarms Active on the QFabric System)

```
user@switch> show chassis alarms
IC-1:
--------------------------------------------------------------------------------
1 alarms currently active
Alarm time               Class  Description
2011-08-24 16:04:15 UTC  Minor  Backup RE Active

Test:
--------------------------------------------------------------------------------
3 alarms currently active
Alarm time               Class  Description
2011-08-24 16:04:15 UTC  Major  Test:fte-0/1/2: Link down
2011-08-24 16:04:14 UTC  Major  Test:fte-0/1/0: Link down
2011-08-24 14:21:14 UTC  Major  Test PEM 0 is not supported/powered

SNG-0:
--------------------------------------------------------------------------------

NW-NG-0:
--------------------------------------------------------------------------------
1 alarms currently active
Alarm time               Class  Description
2011-08-24 15:49:27 UTC  Major  Test PEM 0 is not supported/powered
```

show chassis alarms (Alarms Active on an EX8200 Switch)

```
user@switch> show chassis alarms

6 alarms currently active
Alarm time               Class  Description
2010-12-02 19:15:22 UTC  Major  Fan Tray Failure
2010-12-02 19:15:22 UTC  Major  Fan Tray Failure
2010-12-02 19:15:14 UTC  Minor  Check CB 0 Fabric Chip 1 on Plane/FPC/PFE: 1/5/0,
 1/5/1, 1/5/2, 1/5/3, 1/7/0, 1/7/1, 1/7/2, 1/7/3, 2/5/0, 2/5/1, ...
2010-12-02 19:15:14 UTC  Minor  Check CB 0 Fabric Chip 0 on Plane/FPC/PFE: 1/5/0,
 1/5/1, 1/5/2, 1/5/3, 1/7/0, 1/7/1, 1/7/2, 1/7/3, 2/5/0, 2/5/1, ...
2010-12-02 19:14:18 UTC  Major  PSU 1 Output Failure
2010-12-02 19:14:18 UTC  Minor  Loss of communication with Backup RE
```

show chassis alarms (EX9251 Switch)

```
user@switch> show chassis alarms
 2 alarms currently active
Alarm time               Class  Description
2018-03-08 05:13:10 PST  Major  PEM 0 Not Powered
2018-03-08 05:13:10 PST  Major  Fan Tray 2 is not present
```

show chassis alarms (EX9253 Switch)

```
user@switch> show chassis alarms
6 alarms currently active
Alarm time               Class  Description
2018-03-07 01:09:01 PST  Major  Power Budget:Insufficient Power
2018-03-06 23:56:34 PST  Minor  Loss of communication with Backup RE
2018-02-15 00:48:10 PST  Minor  PEM 3 Not Present
2018-02-15 00:48:10 PST  Minor  PEM 2 Not Present
2018-02-15 00:48:07 PST  Major  PEM 4 Not Powered
2018-02-15 00:48:07 PST  Major  PEM 1 Not Powered
```

show chassis alarms (Alarms Active on a PTX5000 Packet Transport Router)

```
user@host> show chassis alarms

23 alarms currently active
Alarm time               Class  Description
2011-07-12 16:22:05 PDT  Minor  No Redundant Power for Rear Chassis
2011-07-12 16:22:05 PDT  Major  PDU 0 PSM 1 Not OK
2011-07-12 16:21:57 PDT  Minor  No Redundant Power for Fan 0-2
2011-07-12 16:21:57 PDT  Major  PDU 0 PSM 0 Not OK
2011-07-12 15:56:06 PDT  Major  PDU 1 PSM 2 Not OK
2011-07-12 15:56:06 PDT  Minor  No Redundant Power for FPC 0-7
2011-07-12 15:56:06 PDT  Major  PDU 0 PSM 3 Not OK
2011-07-12 15:28:20 PDT  Major  PDU 0 PSM 2 Not OK
2011-07-12 15:19:14 PDT  Minor  Backup RE Active
```

show chassis alarms (Mix of PDUs Alarm on a PTX5000 Packet Transport Router with FPC2-PTX-P1A)

All PDUs installed on a PTX5000 router must be of the same type. The **Mix of PDUs** or **Power Manager Non Operational** alarm is raised when different types of PDUs are installed on a PTX5000 router.

```
user@host> show chassis alarms
15 alarms currently active
Alarm time               Class  Description
2013-03-19 23:03:53 PDT  Minor  No Redundant Power
2013-03-19 23:03:48 PDT  Minor  Mix of PDUs
2013-03-19 23:03:47 PDT  Minor  PDU 1 PSM 3 Absent
2013-03-19 23:03:47 PDT  Minor  PDU 1 PSM 2 Absent
2013-03-19 23:03:47 PDT  Minor  PDU 1 PSM 1 Absent
2013-03-19 23:03:47 PDT  Minor  PDU 1 PSM 0 Absent
2013-03-19 23:03:46 PDT  Major  No CG Online
```

show chassis alarms (PDU Converter Failed Alarm on a PTX5000 Packet Transport Router with FPC2-PTX-P1A)

The **PDU Converter Failed** alarm is raised when one or more 36 V booster converter of a DC PDU fails. If two or more 36 V booster converter fails, fan trays fail and the router might get over heated. Therefore, when this alarm is raised, check the PDU and replace it, if required.

```
user@host> show chassis alarms
```

```
11 alarms currently active
Alarm time               Class  Description
2013-12-11 22:14:13 PST  Minor  No Redundant Power for System
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 7 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 6 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 5 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 4 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 3 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 2 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 1 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 PSM 0 Not OK
2013-12-11 22:14:10 PST  Major  PDU 0 Not OK
2013-12-11 22:14:01 PST  Major  PDU 0 Converter Failed
```

show chassis alarms (No Power for System Alarm on a PTX5000 Packet Transport Router with FPC2-PTX-P1A)

```
user@host> show chassis alarms
```

```
8 alarms currently active
Alarm time               Class  Description
2013-11-19 01:58:41 PST  Major  No Power for System
2013-11-19 01:58:37 PST  Major  PDU 0 PSM 1 Not OK
2013-11-19 01:56:46 PST  Major  PDU 0 PSM 2 Not OK
2013-11-19 01:54:26 PST  Major  PDU 0 PSM 3 Not OK
2013-11-19 01:53:30 PST  Major  PDU 1 PSM 3 Not OK
2013-11-19 01:53:29 PST  Major  PDU 1 PSM 2 Not OK
2013-11-19 01:53:29 PST  Major  PDU 1 PSM 1 Not OK
2013-11-19 01:53:29 PST  Major  PDU 1 PSM 0 Not OK
```

show chassis alarms (Alarms Active on an ACX2000 Universal Metro Router)

```
user@host> show chassis alarms
```

```
7 alarms currently active
Alarm time               Class  Description
2012-05-22 11:19:09 UTC  Major  xe-0/3/1: Link down
2012-05-22 11:19:09 UTC  Major  xe-0/3/0: Link down
2012-05-22 11:19:09 UTC  Major  ge-0/1/7: Link down
2012-05-22 11:19:09 UTC  Major  ge-0/1/6: Link down
2012-05-22 11:19:09 UTC  Major  ge-0/1/3: Link down
2012-05-22 11:19:09 UTC  Major  ge-0/1/2: Link down
2012-05-22 11:19:09 UTC  Major  ge-0/1/1: Link down
```

show chassis alarms (Active Alarm to Indicate Status of the Bad SCB Clock on MX Series)

```
user@host> show chassis alarms
```

```
1 alarm currently active
Alarm time               Class  Description
2013-08-06 07:48:35 PDT  Major  CB 0 19.44 MHz clock failure
```

show chassis alarms (Alarms active on a PTX1000 Packet Transport Router)

```
user@host> show chassis alarms
```

```
2 alarms currently active
Alarm time               Class  Description
2004-08-10 00:55:49 UTC  Major  PEM 1 Not Present
2004-08-10 00:55:49 UTC  Major  PEM 0 Not Present
```

show chassis alarms (MX10003 Router)

If LCMD is down on the backup RE, then the following alarm is seen on the Master.

```
user@host> show chassis alarms
1 alarm currently active
Alarm time               Class  Description
2017-05-09 13:26:27 PDT  Major  VMHost RE 1 host application failed
```

If LCMD is down on the master, then following alarms are displayed.

```
user@host> show chassis alarms
3 alarms currently active
Alarm time               Class  Description
2017-05-10 14:12:21 PDT  Major  VMHost RE 0 host application failed
2017-05-10 14:12:16 PDT  Minor  LCM Peer Absent
2017-05-09 13:26:27 PDT  Major  VMHost RE 1 host application failed
```

If the LCMD process is crashing on the master, the system will switchover after one minute provided the backup RE LCMD connection is stable. The system will not switchover under the following conditions: if the backup RE LCMD connection is unstable or if the current master just gained mastership. When the master has just gained mastership, the switchover happens only after four minutes.

The LCM peer connection un-stable alarm is raised when the LCMD-CHASD IPC communication flaps three times within a small interval of two to three minutes. Once LCM peer connection un-stable alarm is raised, the connection status is monitored for two minutes.

```
user@host> show chassis alarms
7 alarms currently active
Alarm time               Class  Description
2017-05-29 10:12:17 PDT  Minor  LCM Peer Connection un-stable
2017-05-29 09:04:17 PDT  Minor  PEM 8 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 9 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 7 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 3 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 0 Not Powered
2017-05-29 09:04:08 PDT  Minor  Loss of communication with Backup RE
```

If there are no more connection flaps within this two minutes time interval, the LCM peer connection un-stable alarm is cleared.

```
6 alarms currently active
Alarm time               Class  Description
2017-05-29 09:04:17 PDT  Minor  PEM 8 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 9 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 7 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 3 Not Powered
2017-05-29 09:04:17 PDT  Minor  PEM 0 Not Powered
2017-05-29 09:04:08 PDT  Minor  Loss of communication with Backup RE
```

A major alarm is raised even if there is on one PLL lock error, and this alarm can be cleared only through an FPC restart.

```
user@host> show chassis alarms
4 alarms currently active
Alarm time               Class  Description
2017-02-16 09:06:06 PDT  Major  FPC 0 Major Errors
2017-02-16 09:08:40 PDT  Major  FPC 1 Major Errors
2017-02-16 09:11:47 PST  Minor  Fan Tray 3 Pair 1 Outer Fan running at over speed
2017-02-16 09:11:47 PST  Minor  Fan Tray 3 Pair 1 Inner Fan running at over speed
```

show chassis alarms (Alarms active on a MX10008 Router)

```
user@host> show chassis alarms
13 alarms currently active
Alarm time               Class  Description
2018-07-17 05:48:08 PDT  Major  FPC 2 I2C Failure
2018-07-17 05:47:02 PDT  Minor  Mixed Master and Backup RE types
2018-07-17 05:47:01 PDT  Major  Fan Tray 0 Fan 5 Failed
2018-07-17 05:47:01 PDT  Major  Fan Tray 0 Fan 4 Failed
2018-07-17 05:47:01 PDT  Minor  PEM 5 Not Powered
2018-07-17 05:47:01 PDT  Minor  PEM 5 Feed 2 has no input source
2018-07-17 05:47:01 PDT  Minor  PEM 5 Feed 1 has no input source
2018-07-17 05:47:01 PDT  Minor  PEM 4 Not Powered
2018-07-17 05:47:01 PDT  Minor  PEM 4 Feed 2 has no input source
2018-07-17 05:47:01 PDT  Minor  PEM 4 Feed 1 has no input source
2018-07-17 05:47:01 PDT  Minor  PEM 3 Not Powered
2018-07-17 05:47:01 PDT  Minor  PEM 3 Feed 2 has no input source
2018-07-17 05:47:01 PDT  Minor  PEM 3 Feed 1 has no input source
```

show system errors active

Syntax

```
show system errors active
<detail [fru slot-number [scope error-scope] [category error-category]]>
<fru slot-number>
```

Release Information Command introduced in Junos OS Release 18.2R1.

Description Display information collected by the J-Insight fault monitoring feature. Specifically, display summary or detailed information about the active errors based on FRU, error scope, or error category.

Options

none—Display a brief summary of the system error information for all applicable FRUs.

category *error-category*—(Optional) Display system error information based on error category. An error category categorizes errors into various subgroups under a specific error scope level. Values include: core, functional, io, memory, processing, storage, and switch.

detail—(Optional) Display detailed system error information.

fru slot-number—(Optional) Display system error information for a specific FRU. FRU options include fpc, re, and sib.

scope *error-scope*—(Optional) Display system error information based on error scope. An error scope provides a level of classification above error category. Values include: board, pfe, and scope-all.

Required Privilege Level admin

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- show system errors count on page 337
- show system errors error-id on page 339
- show system errors fru on page 341

List of Sample Output

show system errors active on page 333
show system errors active fpc-slot on page 335
show system errors active detail on page 335

Output Fields Table 14 on page 333 list the output fields for the **show system errors active** command. Output fields are listed in the approximate order in which they appear.

Table 14: show system errors active Output Fields

Field Name	Field Description
Error Name	Name of error.
Identifier	Each error is uniquely identified with an error ID that is represented as a Uniform Resource Identifier (URI).
Description	Description of the error.
State	State of the error. Values are: enabled or disabled.
Scope	Scope classification to which the error belongs. Values include board and pfe.
Category	Category subgroup under the scope level to which the error belongs. Values include: core, functional, io, memory, processing, storage, and switch.
Level	Severity level of the error.
Threshold	Configured threshold value. The associated detection and recovery actions are triggered when this value is exceeded.
Error Limit	The maximum number of times the error is reported.
Support	Support details for the error type.
Occur count	Number of times errors of a specific scope, category, and severity level has occurred.
Clear count	Number of times error instances have been cleared.
Last occurred (ms ago)	Amount of time (in milliseconds) passed since the error last occurred.

Sample Output

show system errors active

```
user@host> show system errors active

System Active Errors Information
CB 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
CHASSIS 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 5
Active Fatal Errors           : 0
FAN 0
```

```
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FAN 1
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FAN 2
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FAN 3
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FAN 4
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FPC 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FPC 1
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FPC 2
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FPC 3
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
FPM 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PDU 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PICS 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PICS 1
```

```
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PSM 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PSM 1
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PSM 2
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
PSM 3
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
RE 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
SIB 0
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
SIB 1
---------------------------------
Active Minor Errors           : 0
Active Major Errors           : 0
Active Fatal Errors           : 0
```

show system errors active fpc-slot

```
user@host> show system errors active fpc-slot 0
```

```
System Active Errors Information
FPC 0
-------------------------------------------------
Active Minor Errors: 0
Active Major Errors: 1
Active Fatal Errors: 0
```

show system errors active detail

```
user@host> show system errors active detail
```

```
System Active Errors Detail Information
CHASSIS 0
-----------------------------------------------------------------
Error Name : fan_tray_removal
```

```
Identifier : /chassis/0/hwdre/0/cm/0/fan_tray/Fan Tray 0/fan_tray_removal
Description : Fan_tray_absent
State : disabled
Scope : board
Category : functional
Level : major
Threshold : 1
Error limit : 1
Support : No help info provided
Occur count : 1
Clear count : 0
Last occurred(ms ago) : 339112691
```

show system errors count

Syntax **show system errors count**

Release Information Command introduced in Junos OS Release 18.2R1.

Description Display information collected by the J-Insight fault monitoring feature. Specifically, display information about the number of detected errors and recovery actions triggered based on error severity level.

Options This command has no options.

Required Privilege Level admin

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- show system errors active on page 332
- show system errors error-id on page 339
- show system errors fru on page 341

List of Sample Output show system errors count on page 337

Output Fields Table 15 on page 337 lists the output fields for the **show system errors count** command. Output fields are listed in the approximate order in which they appear.

Table 15: show system errors count Output Fields

Field Name	Field Description
Level	Severity level of the error. Values are: Minor, Major, or Fatal.
Occurred	Number of times errors of a specific severity level occurred.
Cleared	Number of times errors of a specific severity level were cleared.
Action-Taken	Number of times a recovery action was triggered for a specific severity level.

Sample Output

show system errors count

```
user@host> show system errors count
Level   Occurred   Cleared      Action-Taken
------------------------------------------------------------
```

```
Minor:  0          0            0
Major:  1          0            1
Fatal:  0          0            0
```

show system errors error-id

Syntax **show system errors error-id** *error-id-uri*

Release Information Command introduced in Junos OS Release 19.1R1.

Description Display information collected by the J-Insight fault monitoring feature. Specifically, display information about detected errors based on the error ID Uniform Resource Identifier (URI). Only the ouput for errors that have occurred at least once in the system are displayed.

Options This command has no options.

Additional Information

Required Privilege Level admin

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- show system errors active on page 332
- show system errors count on page 337
- show system errors fru on page 341

List of Sample Output show system errors error-id on page 340

Output Fields Table 16 on page 339 lists the output fields for the **show system errors error-id** command. Output fields are listed in the approximate order in which they appear.

Table 16: show system errors error-id Output Fields

Field Name	Field Description
Error Name	Name of error.
Identifier	Each error is uniquely identified with an error ID that is represented as a Uniform Resource Identifier (URI).
Description	Description of the error.
State	State of the error. Values are: enabled or disabled.
Scope	Scope classification to which the error belongs. Values include board and pfe.
Category	Category subgroup under the scope level to which the error belongs. Values include: core, functional, io, memory, processing, storage, and switch.

Table 16: show system errors error-id Output Fields (continued)

Field Name	Field Description
Level	Severity level of the error.
Threshold	Configured threshold value. The associated detection and recovery actions are triggered when this value is exceeded.
Error Limit	The maximum number of times the error is reported.
Support	Support details for the error type.
Occur count	Number of times errors of a specific scope, category, and severity level has occurred.
Clear count	Number of times error instances have been cleared.
Last occurred (ms ago)	Amount of time (in milliseconds) passed since the error last occurred.

Sample Output

show system errors error-id

```
user@host> show system errors error-id "/chassis/0/hwdre/0/cm/0/fan_tray/Fan Tray
0/fan_tray_removal"
System Errors Detail Information
CHASSIS 0
-------------------------------------------------------------------
Error Name            : fan_tray_removal
Identifier            : /chassis/0/hwdre/0/cm/0/fan_tray/Fan Tray
0/fan_tray_removal
Description           : Fan_tray_absent
State                 : enabled
Scope                 : board
Category              : functional
Level                 : major
Threshold             : 1
Error limit           : 1
Support               : No help info provided
Occur count           : 1
Clear count           : 0
Last occurred(ms ago) : 84091182
```

show system errors fru

Syntax **show system errors fru detail [*fru slot-number*]**

Release Information Command introduced in Junos OS Release 18.2R1.

Description Display information collected by the J-Insight fault monitoring feature. Specifically, display information about detected errors based on the FRU.

Options **none**—Display a brief summary of the system error information for the FRU.

detail—(Optional) Display detailed system error information.

fru slot-number—(Optional) Display system error information for a specific FRU. FRU options include fpc, re, and sib.

Required Privilege Level admin

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- show system errors active on page 332
- show system errors count on page 337
- show system errors error-id on page 339

List of Sample Output

show system errors fru detail on page 342
show system errors fru detail (Junos OS Evolved) on page 360

Output Fields Table 17 on page 341 lists the output fields for the **show system errors fru** command. Output fields are listed in the approximate order in which they appear.

Table 17: show system errors fru Output Fields

Field Name	Field Description
FRU	FRU identification number.
Scope	An error scope provides a level of classification above error category. Error scope values are: pfe and board.
Category	An error category categorizes errors into various subgroups under a specific error scope level. Values include: functional, io, memory, processing, storage, and switch.
Level	Severity level of the error.

Table 17: show system errors fru Output Fields (continued)

Field Name	Field Description
Occurred	Number of times errors of a specific scope, category, and severity level has occurred.
Cleared	Number of times errors of a specific scope, category, and severity level were cleared.
Threshold	Configured threshold value. The associated detection and recovery actions are triggered when this value is exceeded.
Action-Taken	Number of times a user-configured recovery action was triggered for errors of a specific scope, category, and severity level .
Action	Action that is triggered when the threshold value is exceeded.

Sample Output

show system errors fru detail

```
user@host> show system errors fru detail

Fru     Scope   Category    Level   Occurred    Cleared     Threshold
Action-Taken    Action
CB 0
        board
                functional  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                      io    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                 memory     Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
               processing   Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                 storage    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
```

```
                          Fatal        0          0           1          0
       CM ALARM|RESET
                 switch   Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
CHASSIS 0
       board
               functional Minor        0          0          10          0
         LOG|
                          Major        5          0           1         10
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
                     io   Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
                 memory   Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
               processing Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
                 storage  Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
                 switch   Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
FAN 0
       board
               functional Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
                     io   Minor        0          0          10          0
         LOG|
                          Major        0          0           1          0
       GET STATE|CM ALARM|
                          Fatal        0          0           1          0
       CM ALARM|RESET
```

```
                  memory    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                 processing  Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                  storage    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                   switch    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
FAN 1
      board
                 functional  Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                       io    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                  memory    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                 processing  Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                  storage    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
                            Fatal       0           0           1          0
      CM ALARM|RESET
                   switch    Minor       0           0          10          0
        LOG|
                            Major       0           0           1          0
      GET STATE|CM ALARM|
```

```
                          Fatal        0          0           1          0
        CM ALARM|RESET
FAN 2
        board
                functional  Minor      0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                     io   Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                 memory   Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                processing Minor       0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                 storage  Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                  switch  Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
FAN 3
        board
                functional  Minor      0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                     io   Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
                 memory   Minor        0          0          10          0
          LOG|
                          Major        0          0           1          0
        GET STATE|CM ALARM|
                          Fatal        0          0           1          0
        CM ALARM|RESET
```

```
               processing  Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                storage    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                 switch    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
FAN 4
        board
               functional  Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                     io    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                 memory    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
               processing  Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                storage    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
                 switch    Minor        0           0          10          0
          LOG|
                           Major        0           0           1          0
        GET STATE|CM ALARM|
                           Fatal        0           0           1          0
        CM ALARM|RESET
FPC 0
        board
               functional  Minor        0           0          10          0
          LOG|
```



```
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
              io    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
          memory    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
        processing  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
         storage    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
          switch    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
pfe
        functional  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
              io    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
          memory    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
        processing  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
GET STATE|CM ALARM|
                    Fatal       0           0           1           0
DISABLE PFE
         storage    Minor       0           0          10           0
```

```
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                    switch    Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
FPC 1
       board
              functional      Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                      io      Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                    memory    Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
              processing      Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                   storage    Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                    switch    Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
       pfe
              functional      Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                      io      Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
       GET STATE|CM ALARM|
```

```
                          Fatal       0           0            1          0
       DISABLE PFE
                 memory   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
               processing Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                storage   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                 switch   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
FPC 2
       board
               functional Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                     io   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                 memory   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
               processing Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                storage   Minor       0           0           10          0
         LOG|
                          Major       0           0            1          0
       GET STATE|CM ALARM|
                          Fatal       0           0            1          0
       DISABLE PFE
                 switch   Minor       0           0           10          0
         LOG|
```

```
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
      pfe
              functional  Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
                    io    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
                memory    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
              processing  Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
               storage    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
                switch    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
FPC 3
      board
              functional  Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
                    io    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      DISABLE PFE
                memory    Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
```

```
       DISABLE PFE
              processing  Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
               storage    Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
                switch    Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
       pfe
              functional  Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
                     io   Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
                memory    Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
              processing  Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
               storage    Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
                switch    Minor       0          0          10         0
         LOG|
                          Major       0          0           1         0
       GET STATE|CM ALARM|
                          Fatal       0          0           1         0
       DISABLE PFE
FPM 0
       board
              functional  Minor       0          0          10         0
         LOG|
```

```
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                    io    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                memory    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
              processing  Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
               storage    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                switch    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
PDU 0
       board
              functional  Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                    io    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                memory    Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
              processing  Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
```

```
                storage    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                 switch    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
PICS 0
        board
               functional  Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                     io    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                 memory    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
               processing  Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                storage    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                 switch    Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
PICS 1
        board
               functional  Minor        0            0           10          0
          LOG|
                           Major        0            0            1          0
        GET STATE|CM ALARM|
                           Fatal        0            0            1          0
        CM ALARM|RESET
                     io    Minor        0            0           10          0
          LOG|
```

```
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                 memory   Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                processing Minor      0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                 storage  Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                 switch   Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
PSM 0
       board
                functional Minor      0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                     io   Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                 memory   Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                processing Minor      0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
                 storage  Minor       0           0          10           0
         LOG|
                          Major       0           0           1           0
       GET STATE|CM ALARM|
                          Fatal       0           0           1           0
       CM ALARM|RESET
```

```
                  switch    Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
PSM 1
        board
                functional  Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                     io     Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                  memory    Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                processing  Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                 storage    Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                  switch    Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
PSM 2
        board
                functional  Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                     io     Minor       0           0           10          0
          LOG|
                            Major       0           0           1           0
        GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        CM ALARM|RESET
                  memory    Minor       0           0           10          0
          LOG|
```

```
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
             processing   Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
              storage     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
               switch     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
PSM 3
      board
             functional   Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
                   io     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
               memory     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
             processing   Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
              storage     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
               switch     Minor       0           0          10           0
        LOG|
                          Major       0           0           1           0
      GET STATE|CM ALARM|
                          Fatal       0           0           1           0
      CM ALARM|RESET
```

```
RE 0
        board
                functional  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                      io    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                  memory    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                processing  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                 storage    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                  switch    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
SIB 0
        board
                functional  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
         RESET
                      io    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
         RESET
                  memory    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
         RESET
                processing  Minor        0           0           10          0
          LOG|
```

```
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                 storage    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                  switch    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
       switch
                functional  Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                      io    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                  memory    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
               processing   Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                 storage    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
                  switch    Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
        RESET
SIB 1
       board
                functional  Minor       0           0          10           0
         LOG|
                            Major       0           0           1           0
       GET STATE|CM ALARM|
                            Fatal       0           0           1           0
```

```
 RESET
              io    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
          memory    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
        processing  Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
         storage    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
          switch    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
switch
        functional  Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
              io    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
          memory    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
        processing  Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
                    Fatal        0           0            1          0
 RESET
         storage    Minor        0           0           10          0
  LOG|
                    Major        0           0            1          0
GET STATE|CM ALARM|
```

```
                          Fatal       0          0          1          0
        RESET
                 switch   Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
        RESET
```

show system errors fru detail (Junos OS Evolved)

```
user@router# show system errors fru detail

Fru     Scope   Category    Level   Occurred    Cleared     Threshold
Action-Taken    Action
CB 0
        board
                functional  Minor       1          0          10         2
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
                     io   Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
                 memory   Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
               processing Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
                storage   Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
                 switch   Minor       0          0          10         0
          LOG|
                          Major       0          0          1          0
       GET STATE|CM ALARM|
                          Fatal       0          0          1          0
       CM ALARM|RESET
CHASSIS 0
        board
                functional  Minor       0          0          10         0
          LOG|
                          Major       1          0          1          2
```

```
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                     io      Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                  memory     Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                processing   Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                 storage     Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                  switch     Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
FAN 1
        board
                functional   Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                     io      Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                  memory     Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                processing   Minor        0          0          10          0
          LOG|
                             Major        0          0           1          0
        GET STATE|CM ALARM|
                             Fatal        0          0           1          0
        CM ALARM|RESET
                 storage     Minor        0          0          10          0
```

```
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                   switch     Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
FAN 2
        board
                 functional   Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                       io     Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                    memory    Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                 processing   Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                  storage     Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                   switch     Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
FAN 3
        board
                 functional   Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
        GET STATE|CM ALARM|
                              Fatal        0            0             1          0
        CM ALARM|RESET
                       io     Minor        0            0            10          0
          LOG|
                              Major        0            0             1          0
```

```
        GET STATE|CM ALARM|
                             Fatal        0           0            1          0
        CM ALARM|RESET
                  memory     Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        GET STATE|CM ALARM|
                             Fatal        0           0            1          0
        CM ALARM|RESET
                processing   Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        GET STATE|CM ALARM|
                             Fatal        0           0            1          0
        CM ALARM|RESET
                  storage    Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        GET STATE|CM ALARM|
                             Fatal        0           0            1          0
        CM ALARM|RESET
                  switch     Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        GET STATE|CM ALARM|
                             Fatal        0           0            1          0
        CM ALARM|RESET
FPC 0
        board
                functional   Minor        2           0           10          4
          LOG|
                             Major       21          21            1         63
        CM ALARM|
                             Fatal        0           0            1          0
        DISABLE PFE
                       io    Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        CM ALARM|
                             Fatal        0           0            1          0
        DISABLE PFE
                  memory     Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        CM ALARM|
                             Fatal        0           0            1          0
        DISABLE PFE
                processing   Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        CM ALARM|
                             Fatal        0           0            1          0
        DISABLE PFE
                  storage    Minor        0           0           10          0
          LOG|
                             Major        0           0            1          0
        CM ALARM|
                             Fatal        0           0            1          0
        DISABLE PFE
                  switch     Minor        0           0           10          0
```

```
          LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
       pfe
              functional      Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                     io       Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                memory        Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
              processing      Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
               storage        Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                switch        Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
FPC 1
       board
              functional      Minor       2           0          10           4
         LOG|
                              Major       1           0           1           2
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                     io       Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
                              Fatal       0           0           1           0
       DISABLE PFE
                memory        Minor       0           0          10           0
         LOG|
                              Major       0           0           1           0
       CM ALARM|
```

```
                          Fatal       0          0           1          0
      DISABLE PFE
             processing   Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
              storage     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
               switch     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
      pfe
             functional   Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
                   io     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
               memory     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
             processing   Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
              storage     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
               switch     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
      DISABLE PFE
FPM 0
      board
             functional   Minor       0          0          10          0
```

```
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                    io     Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                 memory    Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
               processing  Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                storage    Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                 switch    Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
PDU 0
      board
               functional  Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                    io     Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
                 memory    Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
      CM ALARM|RESET
               processing  Minor       0           0          10           0
        LOG|
                           Major       0           0           1           0
      GET STATE|CM ALARM|
                           Fatal       0           0           1           0
```

```
       CM ALARM|RESET
                storage    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                 switch    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
PICS 0
       board
              functional   Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                    io    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                 memory    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
              processing   Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                storage    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                 switch    Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
PICS 1
       board
              functional   Minor        0           0           10          0
         LOG|
                           Major        0           0            1          0
       GET STATE|CM ALARM|
                           Fatal        0           0            1          0
       CM ALARM|RESET
                    io    Minor        0           0           10          0
```

```
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                   memory     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                 processing   Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                  storage     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                   switch     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
PSM 0
        board
                 functional   Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                       io     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                   memory     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                 processing   Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
        CM ALARM|RESET
                  storage     Minor        0           0           10           0
          LOG|
                              Major        0           0            1           0
        GET STATE|CM ALARM|
                              Fatal        0           0            1           0
```

```
        CM ALARM|RESET
                  switch    Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
PSM 1
        board
                functional  Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                     io     Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                  memory    Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                processing  Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                 storage    Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                  switch    Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
PSM 2
        board
                functional  Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                     io     Minor        0            0            10           0
          LOG|
                            Major        0            0             1           0
        GET STATE|CM ALARM|
                            Fatal        0            0             1           0
        CM ALARM|RESET
                  memory    Minor        0            0            10           0
```

```
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                 processing   Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                  storage     Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                   switch     Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
PSM 3
        board
                 functional   Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                        io    Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                   memory     Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                 processing   Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                  storage     Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
        CM ALARM|RESET
                   switch     Minor       0           0          10           0
          LOG|
                              Major       0           0           1           0
        GET STATE|CM ALARM|
                              Fatal       0           0           1           0
```

```
        CM ALARM|RESET
RE 0
        board
                functional  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                      io    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                   memory   Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                processing  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                 storage    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
                  switch    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        GET STATE|CM ALARM|
                            Fatal        0           0            1          0
        CM ALARM|RESET
SIB 0
        board
                functional  Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        CM ALARM|
                            Fatal        0           0            1          0
         RESET
                      io    Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        CM ALARM|
                            Fatal        0           0            1          0
         RESET
                   memory   Minor        0           0           10          0
          LOG|
                            Major        0           0            1          0
        CM ALARM|
                            Fatal        0           0            1          0
         RESET
                processing  Minor        0           0           10          0
```

```
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                     storage  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                      switch  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
       switch
                    functional Minor      0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                          io  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                      memory  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                    processing Minor      0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                     storage  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
                      switch  Minor       0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
                              Fatal       0           0            1           0
        RESET
SIB 1
       board
                    functional Minor      0           0           10           0
          LOG|
                              Major       0           0            1           0
       CM ALARM|
```

```
                    Fatal       0           0           1           0
 RESET
              io    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
          memory    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
        processing  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
         storage    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
          switch    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
switch
        functional  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
              io    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
          memory    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
        processing  Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
CM ALARM|
                    Fatal       0           0           1           0
 RESET
         storage    Minor       0           0          10           0
  LOG|
                    Major       0           0           1           0
```

```
      CM ALARM|
                          Fatal       0          0           1          0
       RESET
               switch     Minor       0          0          10          0
        LOG|
                          Major       0          0           1          0
      CM ALARM|
                          Fatal       0          0           1          0
       RESET
```

Starting in Junos OS Evolved Release 19.1R1, the **show system errors fru detail** output shows error details not just for FPCs but also for other components such as fan, PSM, CB, and chassis.

show system health-monitor

Syntax

show system health-monitor
<fpc fpc-slot *fpc-slot*>

Release Information

Command introduced in Junos OS Release 18.2R1.

Description

Display the J-Insight health monitor results. Starting with Junos OS Release 18.2R1, J-Insight supports health monitoring for FPC FRUs on the MX Series routers.

Options

none—Display information for all FPCs.

fpc fpc-slot *fpc-slot*—(Optional) Display information for a specified FPC.

Required Privilege Level

admin

Related Documentation

- J-Insight Device Monitor Basic Configuration on page 222
- delete services jinsightd subscribe health-monitor on page 303
- set services jinsightd traceoptions on page 311

List of Sample Output

show system health-monitor on page 376

Output Fields

Table 18 on page 375 lists the output fields for the **show system health-monitor** command. Output fields are listed in the approximate order in which they appear.

Table 18: show system health-monitor Output Fields

Field Name	Field Description
Component	Platform component name.
Health-Parameter	Health parameter name.
Value	Reported health value collected by the health monitor.
Threshold	Default threshold value for the health parameter.
Health-Status	State of the health parameter. Values are: GREEN, YELLOW, or RED.
FPC SLOT	FPC slot number.

Sample Output

show system health-monitor

```
user@host> show system health-monitor
```

```
Component      Health-Parameter     Value         Threshold       Health-Status
-------------------------------------------------------------------------------
FPC SLOT: 0

board.0.cpu.0     CPU Load 1 (1  sec)         15              NA          NA
board.0.cpu.0     CPU Load 2 (5  sec)         16              NA          NA
board.0.cpu.0     CPU Load 3 (10 sec)         15              NA          NA
board.0.cpu.0     CPU Load 4 (1  min)         15              NA          NA
board.0.cpu.0     heap_util[Kernel]           11              NA          NA
board.0.cpu.0     heap_util[LAN buffer]       20              NA          NA
board.0.temp.0    Exhaust A                   46 C/114.8 F    75          GREEN
board.0.temp.0    Exhaust B                   59 C/138.2 F    75          GREEN
board.0.temp.0    Intake                      41 C/105.8 F    75          GREEN
board.0.temp.0    LU 0 Chip                   55 C/131 F      NA          NA
board.0.temp.0    LU 0 TSen                   50 C/122 F      NA          NA
board.0.temp.0    LU 1 Chip                   49 C/120.2 F    NA          NA
board.0.temp.0    LU 1 TSen                   50 C/122 F      NA          NA
board.0.temp.0    LU 2 Chip                   57 C/134.6 F    NA          NA
board.0.temp.0    LU 2 TSen                   50 C/122 F      NA          NA
board.0.temp.0    LU 3 Chip                   64 C/147.2 F    NA          NA
board.0.temp.0    LU 3 TSen                   50 C/122 F      NA          NA
board.0.temp.0    PLX Switch Chip             55 C/131 F      NA          NA
board.0.temp.0    PLX Switch TSen             50 C/122 F      NA          NA
board.0.temp.0    XF 0 Chip                   69 C/156.2 F    NA          NA
board.0.temp.0    XF 0 TSen                   50 C/122 F      NA          NA
board.0.temp.0    XM 0 Chip                   58 C/136.4 F    NA          NA
board.0.temp.0    XM 0 TSen                   50 C/122 F      NA          NA
npu.0.fabric.0    PLANE0.dest[0-31]           0x80000003      NA          NA
npu.0.fabric.0    PLANE0.dest[128-159]        0x00000300      NA          NA
npu.0.fabric.0    PLANE0.dest[160-191]        0x20000000      NA          NA
npu.0.fabric.0    PLANE0.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE0.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE0.dest[32-63]          0x00200000      NA          NA
npu.0.fabric.0    PLANE0.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE0.dest[96-127]         0x00000080      NA          NA
npu.0.fabric.0    PLANE1.dest[0-31]           0x80000003      NA          NA
npu.0.fabric.0    PLANE1.dest[128-159]        0x00000300      NA          NA
npu.0.fabric.0    PLANE1.dest[160-191]        0x20000000      NA          NA
npu.0.fabric.0    PLANE1.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE1.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE1.dest[32-63]          0x00200000      NA          NA
npu.0.fabric.0    PLANE1.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE1.dest[96-127]         0x00000080      NA          NA
npu.0.fabric.0    PLANE2.dest[0-31]           0x80000003      NA          NA
npu.0.fabric.0    PLANE2.dest[128-159]        0x00000300      NA          NA
npu.0.fabric.0    PLANE2.dest[160-191]        0x20000000      NA          NA
npu.0.fabric.0    PLANE2.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE2.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE2.dest[32-63]          0x00200000      NA          NA
npu.0.fabric.0    PLANE2.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE2.dest[96-127]         0x00000080      NA          NA
npu.0.fabric.0    PLANE3.dest[0-31]           0x80000003      NA          NA
npu.0.fabric.0    PLANE3.dest[128-159]        0x00000300      NA          NA
npu.0.fabric.0    PLANE3.dest[160-191]        0x20000000      NA          NA
```

```
npu.0.fabric.0    PLANE3.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE3.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE3.dest[32-63]          0x00200000      NA          NA
npu.0.fabric.0    PLANE3.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE3.dest[96-127]         0x00000080      NA          NA
npu.0.fabric.0    PLANE4.dest[0-31]           0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[128-159]        0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[160-191]        0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[32-63]          0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE4.dest[96-127]         0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[0-31]           0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[128-159]        0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[160-191]        0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[32-63]          0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE5.dest[96-127]         0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[0-31]           0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[128-159]        0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[160-191]        0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[32-63]          0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE6.dest[96-127]         0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[0-31]           0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[128-159]        0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[160-191]        0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[192-223]        0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[224-239]        0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[32-63]          0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[64-95]          0x00000000      NA          NA
npu.0.fabric.0    PLANE7.dest[96-127]         0x00000000      NA          NA
npu.0.memory.0    Counters_EDMEM Utilization  50              NA          NA
npu.0.memory.0    EDMEM Utilization           37              NA          NA
npu.0.memory.0    ENCAPS_EDMEM Utilization    100             NA          NA
npu.0.memory.0    Firewall_EDMEM Utilization  1               NA          NA
npu.0.memory.0    HASH_EDMEM Utilization      100             NA          NA
npu.0.memory.0    HASH_OMEM Utilization       100             NA          NA
npu.0.memory.0    IDMEM Utilization           86              NA          NA
npu.0.memory.0    LMEM_LMEM Utilization       100             NA          NA
npu.0.memory.0    Next_Hop_EDMEM Utilization  65              NA          NA
npu.0.memory.0    OMEM Utilization            1               NA          NA
npu.0.memory.0    UEID_SHARED_SPACE_EDMEM Utilization 1       NA          NA
npu.0.memory.0    UEID_SPACE_EDMEM Utilization 1              NA          NA
npu.0.util.0      EDMEM Avg Load              1               NA          NA
npu.0.util.0      Global Utilization          1               NA          NA
npu.0.util.0      IDMEM Avg Load              1               NA          NA
npu.0.util.0      OMEM Avg Load               0               NA          NA
```

show trace

Syntax

```
show trace
<application app-name>
<node node-name>
<pid pid-value>
<time time-elapsed>
```

Release Information Command introduced in Junos OS Evolved Release 18.3R1.

Description Show the trace data from all nodes that is collected on the master Routing Engine in **/var/log/traces**. All applications are traced at the info level for informational messages. You can refine the traces to show by specifying trace time elapsed, application, process ID, and node.

Options

none—Display all traces.

application *app-name*—(Optional) Display traces for the specified application name.

node *node-name*—(Optional) Display traces for the specified node name.

pid *pid-value*—(Optional) Display traces for the specified process ID.

time *time-elapsed*—(Optional) Display traces for the specified elapsed time.

Range: 1 through 840 minutes

Required Privilege Level view

Related Documentation

- clear trace on page 302

List of Sample Output show trace on page 379

Output Fields Table 19 on page 378 lists the output fields for the **show trace** command. Output fields are listed in the approximate order in which they appear.

Table 19: show trace Output Fields

Field Name	Field Description
timestamp	Timestamp field in the following format: YYYY-MM-DD HH:MM:SS.123456789.
node	Node where trace message originated.
system-process	System process where trace message originated.
tracepoint	Tracepoint value of the trace message.

Table 19: show trace Output Fields (continued)

Field Name	Field Description
trace-level	Trace level of the trace message.
application	Application where trace message originated.
message-type	Message type of the trace message.
Function	Function name where the trace message was generated.
Message	Message associated with the tracepoint.

Sample Output

show trace

```
user@host> show trace time 1
2015-06-11 18:35:07.141752545 libevoinfra_INFO_APP  Function =
"evoapp_init_commons", node_type = "RE", node_slot = 0, node_name = "re0", app_name
 = "trace_server", app_id =
0
2015-06-11 18:35:07.141755209 libevoinfra_INFO_2STR  Function =
"evoapp_init_commons", Message1 = "Object subscription mode", Message2 = "Object
 Select"
2015-06-11 18:35:07.141884641 libevoinfra_INFO_2STR  Function = "evoapp_load_dsl",
 Message1 = "App Lua config not set, using app file", Message2 =
"/usr/conf/evoapp/trace_ser
ver.lua"
2015-06-11 18:35:18.418354259 libevoinfra_INFO_STR  Function = "evoapp_zoo_init",
 Message = "Connected to Zookeeper"
2015-06-11 18:35:18.428886547 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "App managed by SysMan", app_name = "trace_server",
 node_name = "re0", app_vers
ion = 0, shared_app_version = 0, node_attr_match = ""
2015-06-11 18:35:18.501195217 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "Generated app local version", app_name =
"trace_server", node_name = "re0", ap
p_version = 5403332628190883088, shared_app_version = 0, node_attr_match = ""
2015-06-11 18:35:18.515955996 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "App shared version", app_name = "trace_server",
node_name = "re0", app_version
 = 5403332628190883088, shared_app_version = 2077664112652750830, node_attr_match
 = ""
2015-06-11 18:35:18.567297706 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "Node attribute match", app_name = "trace_server",
node_name = "re0", app_versi
on = 5403332628190883088, shared_app_version = 2077664112652750830, node_attr_match
 = "RE"
2015-06-11 18:35:18.613447026 libevoinfra_INFO_GUID_BLOCK  Function =
"evoapp_zoo_init", seed = 0x65, start_guid = 433791696896, end_guid = 438086664191,
 total = 4294967295
2015-06-11 18:35:18.613480188 libevoinfra_INFO_STR  Function = "evoapp_init_utils",
 Message = "Enabling AMM by default.."
2015-06-11 18:35:18.777497690 EvoAppService_INFO_IP  Function = "EvoAppService",
```

```
 Message = "Service IP set to", IP = "128.0.0.4"
2015-06-11 18:35:18.843749755 LttngWrapper_INFO  message = "!!!Starting
trace_server!!!

root@evovptxq_RE0-re0> show trace time 1
2015-06-11 18:35:07.141752545 libevoinfra_INFO_APP  Function =
"evoapp_init_commons", node_type = "RE", node_slot = 0, node_name = "re0", app_name
 = "trace_server", app_id =
0
2015-06-11 18:35:07.141755209 libevoinfra_INFO_2STR  Function =
"evoapp_init_commons", Message1 = "Object subscription mode", Message2 = "Object
 Select"
2015-06-11 18:35:07.141884641 libevoinfra_INFO_2STR  Function = "evoapp_load_dsl",
 Message1 = "App Lua config not set, using app file", Message2 =
"/usr/conf/evoapp/trace_ser
ver.lua"
2015-06-11 18:35:18.418354259 libevoinfra_INFO_STR  Function = "evoapp_zoo_init",
 Message = "Connected to Zookeeper"
2015-06-11 18:35:18.428886547 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "App managed by SysMan", app_name = "trace_server",
 node_name = "re0", app_vers
ion = 0, shared_app_version = 0, node_attr_match = ""
2015-06-11 18:35:18.501195217 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "Generated app local version", app_name =
"trace_server", node_name = "re0", ap
p_version = 5403332628190883088, shared_app_version = 0, node_attr_match = ""
2015-06-11 18:35:18.515955996 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "App shared version", app_name = "trace_server",
node_name = "re0", app_version
 = 5403332628190883088, shared_app_version = 2077664112652750830, node_attr_match
 = ""
2015-06-11 18:35:18.567297706 libevoinfra_INFO_EVOAPP  Function =
"evoapp_zoo_init", Message = "Node attribute match", app_name = "trace_server",
node_name = "re0", app_versi
on = 5403332628190883088, shared_app_version = 2077664112652750830, node_attr_match
 = "RE"
2015-06-11 18:35:18.613447026 libevoinfra_INFO_GUID_BLOCK  Function =
"evoapp_zoo_init", seed = 0x65, start_guid = 433791696896, end_guid = 438086664191,
 total = 4294967295
2015-06-11 18:35:18.613480188 libevoinfra_INFO_STR  Function = "evoapp_init_utils",
 Message = "Enabling AMM by default.."
2015-06-11 18:35:18.777497690 EvoAppService_INFO_IP  Function = "EvoAppService",
 Message = "Service IP set to", IP = "128.0.0.4"
2015-06-11 18:35:18.843749755 LttngWrapper_INFO  message = "!!!Starting
trace_server!!!
...
```