



Network Management and Monitoring Feature Guide for EX4600 Switches

Release
15.1



Modified: 2016-03-30

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

Network Management and Monitoring Feature Guide for EX4600 Switches
Release 15.1
Copyright © 2016, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

| | | |
|------------------|--|-----------|
| | About the Documentation | xi |
| | Documentation and Release Notes | xi |
| | Supported Platforms | xi |
| | Using the Examples in This Manual | xi |
| | Merging a Full Example | xii |
| | Merging a Snippet | xii |
| | Documentation Conventions | xiii |
| | Documentation Feedback | xv |
| | Requesting Technical Support | xv |
| | Self-Help Online Tools and Resources | xv |
| | Opening a Case with JTAC | xvi |
| Part 1 | Overview | |
| Chapter 1 | Understanding Network Management | 3 |
| | Understanding Device and Network Management Features | 3 |
| | Understanding Tracing and Logging Operations | 6 |
| Chapter 2 | Automation | 9 |
| | How Commit Scripts Work | 9 |
| | Commit Script Input | 10 |
| | Commit Script Output | 11 |
| | Commit Scripts and the Junos OS Commit Model | 12 |
| | Standard Commit Model | 12 |
| | Commit Model with Commit Scripts | 12 |
| | Avoiding Potential Conflicts When Using Multiple Commit Scripts | 14 |
| | Overview of Generating Persistent or Transient Configuration Changes | 15 |
| | Differences Between Persistent and Transient Changes | 16 |
| | Interaction of Configuration Changes and Configuration Groups | 19 |
| | Tag Elements and Templates for Generating Changes | 19 |
| | Required Boilerplate for Commit Scripts | 20 |
| | How Op Scripts Work | 21 |
| | Required Boilerplate for Op Scripts | 22 |
| Chapter 3 | Junos Space | 25 |
| | Understanding Junos Space Support | 25 |

| | | |
|------------------|---|-----------|
| Part 2 | Configuration | |
| Chapter 4 | Configuring Network Analytics | 29 |
| | Network Analytics Overview | 29 |
| | Analytics Feature Overview | 30 |
| | Network Analytics Enhancements Overview | 31 |
| | Summary of CLI Changes | 32 |
| | Understanding Network Analytics Configuration and Status | 36 |
| | Understanding Network Analytics Streaming Data | 38 |
| | Understanding Enhanced Network Analytics Streaming Data | 40 |
| | Google Protocol Buffer (GPB) | 40 |
| | JavaScript Object Notation (JSON) | 43 |
| | Comma-separated Values (CSV) | 43 |
| | Tab-separated Values (TSV) | 43 |
| | Queue Statistics Output for JSON, CSV, and TSV | 44 |
| | Traffic Statistics Output for JSON, CSV, and TSV | 44 |
| | Understanding Enhanced Analytics Local File Output | 45 |
| | Prototype File for the Google Protocol Buffer Stream Format | 47 |
| | Example: Configuring Enhanced Network Analytics Features | 48 |
| | Configuring Queue Monitoring | 58 |
| | Configuring Traffic Monitoring | 60 |
| | Configuring a Local File for Network Analytics Data | 61 |
| | Configuring a Remote Collector for Streaming Analytics Data | 62 |
| Chapter 5 | Configuring sFlow Technology | 65 |
| | Understanding How to Use sFlow Technology for Network Monitoring on a Switch | 65 |
| | Sampling Mechanism and Architecture of sFlow Technology on Switches | 65 |
| | Adaptive Sampling | 67 |
| | sFlow Agent Address Assignment | 68 |
| | sFlow Limitations on Switches | 68 |
| | Configuring sFlow Technology | 70 |
| Chapter 6 | Configuring SNMP | 73 |
| | Understanding the Implementation of SNMP | 74 |
| | Utility MIB | 76 |
| | SNMPv3 Overview | 77 |
| | Minimum SNMPv3 Configuration on a Device Running Junos OS | 78 |
| | Understanding RMON | 79 |
| | RMON Overview | 79 |
| | Alarm Thresholds and Events | 80 |
| | RMON MIB Event, Alarm, Log, and History Control Tables | 81 |
| | Understanding Health Monitoring | 83 |
| | SNMP MIBs Support | 84 |
| | MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 84 |
| | MIBs Supported on QFabric Systems | 94 |

| | | |
|------------------|--|------------|
| | SNMP Traps Support | 100 |
| | SNMP Traps Supported on QFX Series Standalone Switches and QFX Series | |
| | Virtual Chassis | 100 |
| | SNMPv1 Traps | 100 |
| | SNMPv2 Traps | 105 |
| | SNMP Traps Supported on QFabric Systems | 109 |
| | Configuring SNMP | 113 |
| | Configuring the SNMP Community String | 117 |
| | Configuring SNMP Trap Groups | 117 |
| | Adding a Group of Clients to an SNMP Community | 118 |
| | Configuring the Interfaces on Which SNMP Requests Can Be Accepted | 120 |
| | Configuring MIB Views | 120 |
| | Configuring RMON Alarms and Events | 121 |
| | Configuring SNMP | 121 |
| | Configuring an Event | 122 |
| | Configuring an Alarm | 122 |
| | Configuring Health Monitoring | 123 |
| | Creating SNMPv3 Users | 124 |
| | Configuring Access Privileges for a Group | 125 |
| | Assigning a Security Name to a Group | 126 |
| | Configuring SNMPv3 Traps on a Device Running Junos OS | 127 |
| | Configuring SNMP Informs | 128 |
| Chapter 7 | Configuring System Logging | 131 |
| | Overview of Junos OS System Log Messages | 132 |
| | Overview of Single-Chassis System Logging Configuration | 132 |
| | Examples: Configuring System Logging | 134 |
| | Examples: Assigning an Alternative Facility | 136 |
| | Junos OS Minimum System Logging Configuration | 137 |
| | Junos OS System Log Configuration Statements | 137 |
| | Adding a Text String to System Log Messages | 138 |
| | Directing System Log Messages to a Log File | 139 |
| | Directing System Log Messages to a Remote Machine | 140 |
| | Directing System Log Messages to a User Terminal | 140 |
| | Directing System Log Messages to the Console | 141 |
| | Disabling the System Logging of a Facility | 141 |
| | Displaying a Log File from a Single-Chassis System | 142 |
| | Including Priority Information in System Log Messages | 143 |
| | Including the Year or Millisecond in Timestamps | 144 |
| | Logging Messages in Structured-Data Format | 145 |
| | Interpreting Messages Generated in Structured-Data Format | 146 |
| | Interpreting Messages Generated in Standard Format | 149 |
| | Specifying Log File Size, Number, and Archiving Properties | 150 |
| | Specifying the Facility and Severity of Messages to Include in the Log | 151 |
| | Junos OS System Logging Facilities and Message Severity Levels | 152 |
| | System Log Default Facilities for Messages Directed to a Remote | |
| | Destination | 153 |
| | Alternate Facilities for System Log Messages Directed to a Remote | |
| | Destination | 154 |

| | | |
|-------------------|---|------------|
| | Changing the Alternative Facility Name for System Log Messages Directed to a Remote Destination | 155 |
| | Using Regular Expressions to Refine the Set of Logged Messages | 156 |
| Chapter 8 | Configuration Tasks for Network Management | 159 |
| | Configuring Console and Auxiliary Port Properties | 159 |
| | Configuring SSH Service for Remote Access to the Router or Switch | 160 |
| | Configuring the Root Login Through SSH | 161 |
| | Configuring the SSH Protocol Version | 161 |
| | Configuring the Client Alive Mechanism | 161 |
| | Configuring Telnet Service for Remote Access to a Switch | 162 |
| Part 3 | Monitoring and Troubleshooting | |
| Chapter 9 | Monitoring | 165 |
| | Displaying a Log File from a Single-Chassis System | 165 |
| | Monitoring Traffic Through the Router or Switch | 166 |
| | Displaying Real-Time Statistics About All Interfaces on the Router or Switch | 166 |
| | Displaying Real-Time Statistics About an Interface on the Router or Switch | 167 |
| | Monitoring RMON MIB Tables | 169 |
| | Monitoring SNMP | 169 |
| | Monitoring System Log Messages | 171 |
| | Pinging Hosts | 172 |
| | Tracing SNMP Activity on a Device Running Junos OS | 173 |
| | Configuring the Number and Size of SNMP Log Files | 174 |
| | Configuring Access to the Log File | 174 |
| | Configuring a Regular Expression for Lines to Be Logged | 175 |
| | Configuring the Trace Operations | 175 |
| | Using the Enterprise-Specific Utility MIB to Enhance SNMP Coverage | 176 |
| | Displaying Commit Script Output | 178 |
| Chapter 10 | Troubleshooting | 181 |
| | Recovering from a Failed Software Installation | 181 |
| | Loading a Previous Configuration File | 183 |
| | Reverting to the Default Factory Configuration | 184 |
| | Reverting to the Rescue Configuration | 184 |
| | Recovering the Root Password | 185 |

List of Figures

| | | |
|------------------|---|-----------|
| Part 1 | Overview | |
| Chapter 2 | Automation | 9 |
| | Figure 1: Commit Script Input and Output | 10 |
| | Figure 2: Standard Commit Model | 12 |
| | Figure 3: Commit Model with Commit Scripts Added | 13 |
| | Figure 4: Configuration Evaluation by Multiple Commit Scripts | 15 |
| | Figure 5: Op Script Input and Output | 22 |
| Part 2 | Configuration | |
| Chapter 6 | Configuring SNMP | 73 |
| | Figure 6: SNMP Communication Flow | 75 |
| | Figure 7: Setting Thresholds | 80 |
| | Figure 8: Inform Request and Response | 129 |

List of Tables

| | | |
|------------------|--|-----------|
| | About the Documentation | xi |
| | Table 1: Notice Icons | xiii |
| | Table 2: Text and Syntax Conventions | xiii |
| Part 1 | Overview | |
| Chapter 1 | Understanding Network Management | 3 |
| | Table 3: Device and Network Management Features on the QFX Series, OCX Series, and EX4600 | 3 |
| Chapter 2 | Automation | 9 |
| | Table 4: Differences Between Persistent and Transient Changes | 17 |
| Part 2 | Configuration | |
| Chapter 4 | Configuring Network Analytics | 29 |
| | Table 5: Network Analytics CLI Changes | 32 |
| | Table 6: Configuration and Status Output in Junos OS Release 13.2X51-D10 and 13.2X50-D15 | 37 |
| | Table 7: Streamed Queue Statistics Data Output Fields | 38 |
| | Table 8: Streamed Traffic Statistics Data Output Fields | 39 |
| | Table 9: GPB Stream Format Message Header Information | 41 |
| | Table 10: Streamed Queue Statistics Data Output Fields | 44 |
| | Table 11: Streamed Traffic Statistics Data Output Fields | 44 |
| | Table 12: Output Fields for Queue Statistics in Local Analytics File | 46 |
| | Table 13: Output Fields for Traffic Statistics in Local Analytics File | 46 |
| Chapter 6 | Configuring SNMP | 73 |
| | Table 14: RMON Event Table | 81 |
| | Table 15: RMON Alarm Table | 81 |
| | Table 16: jnxRmon Alarm Table | 82 |
| | Table 17: RMON History Control Table | 83 |
| | Table 18: Monitored Object Instances | 84 |
| | Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 85 |
| | Table 20: Juniper Networks Enterprise-Specific MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 90 |
| | Table 21: Standard MIBs Supported on QFabric Systems | 94 |
| | Table 22: Juniper Networks Enterprise-Specific MIBs Supported on QFabric Systems | 98 |
| | Table 23: Standard SNMP Version 1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 101 |

| | | |
|------------------|--|------------|
| | Table 24: Enterprise-Specific SNMPv1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 103 |
| | Table 25: Standard SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 105 |
| | Table 26: Enterprise-Specific SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis | 107 |
| | Table 27: Standard SNMPv2 Traps Supported on QFabric Systems | 110 |
| | Table 28: Enterprise-Specific SNMPv2 Traps Supported on QFabric Systems | 111 |
| Chapter 7 | Configuring System Logging | 131 |
| | Table 29: Minimum Configuration Statements for System Logging | 137 |
| | Table 30: Fields in Structured-Data Messages | 146 |
| | Table 31: Facility and Severity Codes in the priority-code Field | 148 |
| | Table 32: Fields in Standard-Format Messages | 149 |
| | Table 33: Junos OS System Logging Facilities | 152 |
| | Table 34: System Log Message Severity Levels | 152 |
| | Table 35: Default Facilities for Messages Directed to a Remote Destination | 153 |
| | Table 36: Facilities for the facility-override Statement | 154 |
| | Table 37: Regular Expression Operators for the match Statement | 157 |
| Part 3 | Monitoring and Troubleshooting | |
| Chapter 9 | Monitoring | 165 |
| | Table 38: Output Control Keys for the monitor interface Command | 168 |
| | Table 39: SNMP Tracing Flags | 175 |
| | Table 40: Commit Script Configuration and Operational Mode Commands | 179 |

About the Documentation

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

Documentation and Release Notes

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

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

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

Supported Platforms

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

- EX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

Table 2 on page xiii 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 |

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

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

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

| Convention | Description | Examples |
|------------------------------|---|--|
| > (bold right angle bracket) | Separates levels in a hierarchy of menu selections. | In the configuration editor hierarchy, select Protocols>Ospf . |

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Understanding Network Management on page 3](#)
- [Automation on page 9](#)
- [Junos Space on page 25](#)

CHAPTER 1

Understanding Network Management

- [Understanding Device and Network Management Features on page 3](#)
- [Understanding Tracing and Logging Operations on page 6](#)

Understanding Device and Network Management Features

After you install a QFX Series product, OCX Series device, or EX4600 switch in your network, you need to manage the device. The products support features that you use to manage the device within the network, including the management of configuration, system performance, fault monitoring, and remote access.

[Table 3 on page 3](#) lists the device and network management features on the QFX Series, OCX Series, and EX4600.

Table 3: Device and Network Management Features on the QFX Series, OCX Series, and EX4600

| Feature | Typical Uses | Documentation |
|--|------------------------|---|
| AI-Scripts and Advanced Insight Manager (AIM)—Automatically detect and monitor faults on the switch, and depending on the configuration on the AIM application, send notifications of potential problems, and submit problem reports to Juniper Support Systems. | Fault management | Advanced Insight Scripts (AI-Scripts) Release Notes |
| Alarms and LEDs on the switch—Show status of hardware components and indicate warning or error conditions. | Fault management | Chassis Alarm Messages on a QFX3500 Device |
| Firewall filters—Control the packets that are sent to and from the network, balance network traffic, and optimize performance. | Performance management | <ul style="list-style-type: none">• Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices• Overview of Firewall Filters |

Table 3: Device and Network Management Features on the QFX Series, OCX Series, and EX4600 (*continued*)

| Feature | Typical Uses | Documentation |
|--|--|---|
| In-band management—Enables connection to the switch using the same interfaces through which customer traffic flows. Communication between the switch and a remote console is typically enabled using SSH and Telnet services. SSH provides secure encrypted communications, whereas Telnet provides unencrypted, and therefore less secure, access to the switch. | Remote access management | <ul style="list-style-type: none"> • Configuring SSH Service for Remote Access to the Router or Switch on page 160 • Configuring Telnet Service for Remote Access to a Router or Switch |
| Juniper Networks Junos OS automation scripts—Configuration and operations automation tools provided by Junos OS. These tools include commit scripts, operation scripts, event scripts, and event policies. Commit scripts enforce custom configuration rules, whereas operation scripts, event policies, and event scripts automate network troubleshooting and management. | <ul style="list-style-type: none"> • Configuration management • Performance management • Fault management | <i>Automation Scripting Feature Guide</i> |
| Junos OS command-line interface (CLI)—CLI configuration statements that enable you to configure the switch based on your networking requirements, such as security, service, and performance. | <ul style="list-style-type: none"> • Configuration management • Performance management • User access management • Remote access management | <i>CLI User Guide</i> |
| Junos Space software—Multipurpose GUI-based network management system that includes a base platform, the Network Application Platform, and other optional applications such as Ethernet Design, Service Now, Service Insight, and Virtual Control. NOTE: Junos Space does not support the OCX Series. | <ul style="list-style-type: none"> • Configuration management • Performance management • Fault management | <ul style="list-style-type: none"> • Understanding Junos Space Support on page 25 • Junos Space Network Application Platform User Guide |
| Junos XML API—XML representation of Junos OS configuration statements and operational mode commands. Junos XML configuration tag elements are the content to which the Junos XML protocol operations apply. Junos XML operational tag elements are equivalent in function to operational mode commands in the CLI, which you can use to retrieve status information for a device. The Junos XML API also includes tag elements that are the counterpart to Junos CLI configuration statements. | <ul style="list-style-type: none"> • Configuration management • Performance management • Fault management | <ul style="list-style-type: none"> • Junos XML API Configuration Developer Reference • Junos XML API Operational Developer Reference |

Table 3: Device and Network Management Features on the QFX Series, OCX Series, and EX4600 (*continued*)

| Feature | Typical Uses | Documentation |
|---|--|--|
| NETCONF XML management protocol—XML-based management protocol that client applications use to request and change configuration information on routing, switching, and security platforms running Junos OS. The NETCONF XML management protocol defines basic operations that are equivalent to Junos OS CLI configuration mode commands. Client applications use the protocol operations to display, edit, and commit configuration statements (among other operations), just as administrators use CLI configuration mode commands such as show , set , and commit to perform those operations. | <ul style="list-style-type: none"> • Configuration management • Performance management • Fault management | <i>NETCONF XML Management Protocol Developer Guide</i> |
| Operational mode commands—May be used to do the following: <ul style="list-style-type: none"> • Monitor switch performance. For example, the show chassis routing-engine command shows the CPU utilization of the Routing Engine. High CPU utilization of the Routing Engine can affect performance of the switch. • View current activity and status of the device or network. For example, you can use the ping command to monitor and diagnose connectivity problems, and the traceroute command to locate points of failure on the network. | <ul style="list-style-type: none"> • Performance management • Fault management | CLI Explorer |
| Out-of-band management—Enables connection to the switch through a management interface. Out-of-band management is supported on two dedicated management Ethernet interfaces as well as on the console and auxiliary ports. The management Ethernet interfaces connect directly to the Routing Engine. No transit traffic is allowed through the interfaces, separating customer and management traffic and ensuring that congestion or failures in the transit network do not affect the management of the switch. | Remote access management | <ul style="list-style-type: none"> • Connecting a QFX3500 Device to a Network for Out-of-Band Management • Connecting a QFX Series Device to a Management Console • Configuring Console and Auxiliary Port Properties on page 159 |

Table 3: Device and Network Management Features on the QFX Series, OCX Series, and EX4600 (*continued*)

| Feature | Typical Uses | Documentation |
|--|--|--|
| SNMP Configuration Management MIB—Provides notification for configuration changes in the form of SNMP traps. Each trap contains the time at which the configuration change was committed, the name of the user who made the change, and the method by which the change was made. A history of the last 32 configuration changes is kept in jnxCmChgEventTable. | Configuration management | <i>SNMP MIBs and Traps Reference</i> |
| SNMP MIBs and traps—Enable the monitoring of network devices from a central location. Use SNMP requests such as get and walk to monitor and view system activity. The QFX3500 switch supports SNMP Version 1 (v1), v2, and v3, and both standard and Juniper Networks enterprise-specific MIBs and traps. | Fault management | <ul style="list-style-type: none"> • <i>SNMP MIBs and Traps Reference</i> • Understanding the Implementation of SNMP on page 74 |
| System log messages—Log details of system and user events, including errors. You can specify the severity and type of system log messages you wish to view or save, and configure the output to be sent to local or remote hosts. | <ul style="list-style-type: none"> • Fault management • User access management | <ul style="list-style-type: none"> • <i>Junos OS System Log Messages Reference</i> • Overview of Junos OS System Log Messages on page 132 • Overview of Single-Chassis System Logging Configuration on page 132 |

Understanding Tracing and Logging Operations

Tracing and logging operations enable you to track events that occur in the switch—both normal operations and error conditions—and to track the packets that are generated by or passed through the switch. The results of tracing and logging operations are placed in files in the **/var/log** directory on the switch.

The Junos OS supports remote tracing for the following processes:

- **chassisd**—Chassis-control process
- **eventd**—Event-processing process
- **cosd**—Class-of-service process

You configure remote tracing by using the **tracing** statement at the **[edit system]** hierarchy level.



NOTE: The **tracing** statement is not supported on the QFX3000 QFabric system.

If you enabled remote tracing but wish to disable it for specific processes on the switch, use the **no-remote-trace** statement at the **[edit process-name traceoptions]** hierarchy level. This feature does not alter local tracing functionality in any way, and logging files are stored on the switch.

Logging operations use a system logging mechanism similar to the UNIX **syslogd** utility to record systemwide, high-level operations, such as interfaces going up or down and users logging in to or out of the switch. You configure these operations by using the **syslog** statement at the **[edit system]** hierarchy level and by using the **options** statement at the **[edit ethernet-switching-options]** hierarchy level.

Tracing operations record more detailed information about the operations of the switch, including packet forwarding and routing information. To configure tracing operations, use the **traceoptions** statement.



NOTE: The **traceoptions** statement is not supported on the QFX3000 QFabric system.

You can define tracing operations in different portions of the switch configuration:

- **SNMP agent activity tracing operations**—Define tracing of the activities of SNMP agents on the switch. You configure SNMP agent activity tracing operations at the **[edit snmp]** hierarchy level.
- **Global switching tracing operations**—Define tracing for all switching operations. You configure global switching tracing operations at the **[edit ethernet-switching-options]** hierarchy level of the configuration.
- **Protocol-specific tracing operations**—Define tracing for a specific routing protocol. You configure protocol-specific tracing operations in the **[edit protocols]** hierarchy when configuring the individual routing protocol. Protocol-specific tracing operations override any equivalent operations that you specify in the global **traceoptions** statement. If there are no equivalent operations, they supplement the global tracing options. If you do not specify any protocol-specific tracing, the routing protocol inherits all the global tracing operations.
- **Tracing operations within individual routing protocol entities**—Some protocols allow you to define more granular tracing operations. For example, in Border Gateway Protocol (BGP), you can configure peer-specific tracing operations. These operations override any equivalent BGP-wide operations or, if there are no equivalents, supplement them. If you do not specify any peer-specific tracing operations, the peers inherit, first, all the BGP-wide tracing operations and, second, the global tracing operations.
- **Interface tracing operations**—Define tracing for individual interfaces and for the interface process itself. You define interface tracing operations at the **[edit interfaces]** hierarchy level of the configuration.
- **Remote tracing**—To enable system-wide remote tracing, configure the **destination-override syslog host** statement at the **[edit system tracing]** hierarchy level. This specifies the remote host running the system log process (syslogd), which collects

the traces. Traces are written to files on the remote host in accordance with the syslogd configuration in `/etc/syslog.conf`. By default, remote tracing is not configured.

To override the system-wide remote tracing configuration for a particular process, include the **no-remote-trace** statement at the `[edit process-name traceoptions]` hierarchy. When **no-remote-trace** is enabled, the process does local tracing.

To collect traces, use the **local0** facility as the selector in the `/etc/syslog.conf` file on the remote host. To separate traces from various processes into different files, include the process name or trace-file name (if it is specified at the `[edit process-name traceoptions file]` hierarchy level) in the Program field in the `/etc/syslog.conf` file. If your system log server supports parsing hostname and program name, then you can separate traces from the various processes.



NOTE: During a commit check, warnings about the traceoptions configuration (for example, mismatch in trace file sizes or number of trace files) are not displayed on the console. However, these warnings are logged in the system log messages when the new configuration is committed.

**Related
Documentation**

- [Overview of Junos OS System Log Messages on page 132](#)

CHAPTER 2

Automation

- [How Commit Scripts Work on page 9](#)
- [Avoiding Potential Conflicts When Using Multiple Commit Scripts on page 14](#)
- [Overview of Generating Persistent or Transient Configuration Changes on page 15](#)
- [Required Boilerplate for Commit Scripts on page 20](#)
- [How Op Scripts Work on page 21](#)
- [Required Boilerplate for Op Scripts on page 22](#)

How Commit Scripts Work

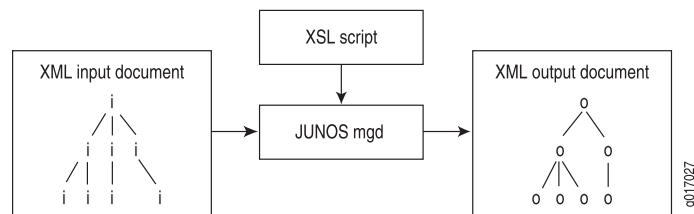
You enable commit scripts by listing the names of one or more commit script files at the **[edit system scripts commit]** hierarchy level. These scripts contain instructions that enforce custom configuration rules. Commit scripts are invoked during the commit process before the standard Junos OS validity checks are performed.

When you perform a commit operation, Junos OS executes each script in turn, passing the information in the candidate configuration to the scripts. The script inspects the configuration, performs the necessary tests and validations, and generates a set of instructions for performing certain actions. These actions include generating error, warning, and system log messages. If errors are generated, the commit operation fails and the candidate configuration remains unchanged. This is the same behavior that occurs with standard commit errors.

Commit scripts can also generate changes to the system configuration. Because the changes are loaded before the standard validation checks are performed, they are validated for correct syntax, just like statements already present in the configuration before the script is applied. If the syntax is correct, the configuration is activated and becomes the active, operational device configuration.

[Figure 1 on page 10](#) shows the flow of commit script input and output.

Figure 1: Commit Script Input and Output



Commit scripts cannot make configuration changes to protected statements or within protected hierarchies. If a commit script attempts to modify or delete a protected statement or hierarchy, Junos OS issues a warning that the change cannot be made. Failure to modify a protected configuration element does not halt the commit script or the commit process.

The following sections discuss several important concepts related to the commit script input and output:

- [Commit Script Input on page 10](#)
- [Commit Script Output on page 11](#)
- [Commit Scripts and the Junos OS Commit Model on page 12](#)

Commit Script Input

The input for a commit script is the postinheritance candidate configuration in Junos XML API format. The term *postinheritance* means that all configuration group values have been inherited by their targets in the candidate configuration and the inactive portions of the configuration have been removed. For more information about configuration groups, see the *CLI User Guide*.

When you issue the **commit** command, Junos OS automatically generates the candidate configuration in XML format and reads it into the management (mgd) process, at which time the input is evaluated by any commit scripts.

To display the XML format of the postinheritance configuration, issue the **show | display commit-scripts view** command:

```
[edit]
user@host# show | display commit-scripts view
```

To display all configuration groups data, including script-generated changes to the groups, issue the **show groups | display commit-scripts** command:

```
[edit]
user@host# show groups | display commit-scripts
```

To save the commit script input to a file, add the **save** command to the command line:

```
[edit]
user@host# show | display commit-scripts view | save filename.xml
```

By default, the file is placed in your home directory on the switch, router, or security device.

Commit Script Output

To specify the desired commit script output—including warning, error, and system log messages, persistent changes, and transient changes—the script can contain tags that appear in any order, in any number. The tags for specifying output are as follows:

- **<xnm:warning>**—Generates a warning message
- **<xnm:error>**—Generates an error message.
- **<syslog><message>**—Generates a system log message.
- **<change>**—Generates a persistent change to the configuration.
- **<transient-change>**—Generates a transient change to the configuration.
- **<xsl:call-template name="jcs:emit-change">**
 <xsl:with-param name="content">—Generates a persistent change relative to the current context node as defined by an XPath expression.
- **<xsl:call-template name="jcs:emit-change">**
 <xsl:with-param name="tag" select="'transient-change'"/>
 <xsl:with-param name="content">—Generates a transient change relative to the current context node as defined by an XPath expression.
- **<xsl:call-template name="jcs:emit-change">**
 <xsl:with-param name="message">
 <xsl:text>—Generates a warning message in conjunction with a configuration change. You can use this set of tags to generate a notification that the configuration has been changed.

Junos OS processes this output and performs the appropriate actions. Errors and warnings are passed back to the Junos OS CLI or to a Junos XML protocol client application. The presence of an error automatically causes the commit operation to fail. Persistent and transient changes are loaded into the appropriate configuration database.

To test the output of error, warning, and system log messages from commit scripts, issue the **commit check | display xml** command:

```
[edit]
user@host# commit check | display xml
```

To display a detailed trace of commit script processing, issue the **commit check | display detail** command:

```
[edit]
user@host# commit check | display detail
```



NOTE: System log messages do not appear in the trace output, so you cannot use the commit check operation to test script-generated system log messages. Furthermore, system log messages are written to the system log during a commit operation, but not during a commit check operation.

- Related Documentation**
- *Example: Protecting the Junos OS Configuration from Modification or Deletion*
 - *jcs:emit-change Template*

Commit Scripts and the Junos OS Commit Model

Junos OS uses a commit model to update the device's configuration. This model allows you to make a series of changes to a candidate configuration without affecting the operation of the device. When the changes are complete, you can commit the configuration. The commit operation saves the candidate configuration changes into the current configuration.

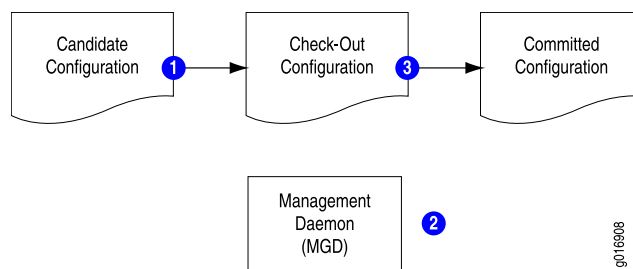
When you commit a set of changes in the candidate configuration, two methods are used to forward these changes to the current configuration:

- Standard commit model—Used when no commit scripts are active on the device.
- Commit script model—Incorporates commit scripts into the commit model.

Standard Commit Model

In the standard commit model, the management (mgd) process validates the candidate configuration based on standard Junos validation rules. If the configuration file is valid, it becomes the current active configuration. [Figure 2 on page 12](#) and the accompanying discussion explain how the standard commit model works:

Figure 2: Standard Commit Model



In the standard commit model, the software performs the following steps:

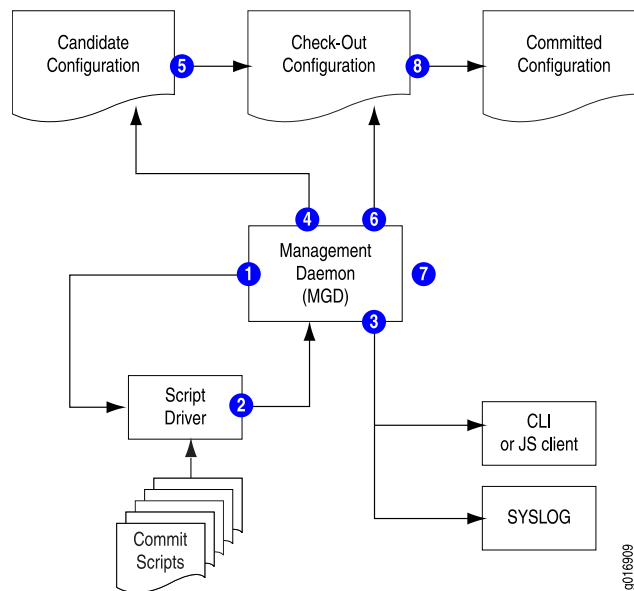
1. When the candidate configuration is committed, it is copied to become the checkout configuration.
2. The mgd process validates the checkout configuration.
3. If no error occurs, the checkout configuration is copied as the current active configuration.

Commit Model with Commit Scripts

When commit scripts are added to the standard commit model, the process becomes more complex. The mgd process first passes an XML-formatted checkout configuration to a script driver, which handles the verification of the checkout configuration by the commit scripts. When verification is complete, the script driver returns an XML *action file*

to the mgd process. The mgd process follows the instructions in the action file to update the candidate and checkout configurations, issue messages to the CLI, and write information to the system log as required. After processing the action file, the mgd process performs the standard Junos OS validation. [Figure 3 on page 13](#) and the accompanying discussion explain this process.

Figure 3: Commit Model with Commit Scripts Added



In the commit script model, Junos OS performs the following steps:

1. When the candidate configuration is committed, the mgd process sends the XML-formatted candidate configuration to the script driver.
2. Each enabled commit script is invoked against the candidate configuration, and each script can generate a set of actions for the mgd process to perform. The actions are collected in an XML action file.
3. The mgd process performs the following actions in response to **<error>**, **<warning>**, and **<syslog>** tag elements in the action file:
 - **<error>**—The mgd process halts the commit process (that is, the commit operation fails), returns an error message to the CLI or Junos XML protocol client, and takes no further action.
 - **<warning>**—The mgd process forwards the message to the CLI or the Junos XML protocol client.
 - **<syslog>**—The mgd process forwards the message to the system log process.
4. If the action file includes any **<change>** tag elements, the mgd process loads the requested changes into the candidate configuration.
5. The candidate configuration is copied to become the checkout configuration.
6. If the action file includes any **<transient-change>** tag elements, the mgd process loads the requested changes into the checkout configuration.

7. The mgd process validates the checkout configuration.
8. If there are no validation errors, the checkout configuration is copied to become the current active configuration.



NOTE: Commit scripts cannot make configuration changes to protected statements or within protected hierarchies. If a commit script attempts to modify or delete a protected statement or hierarchy, Junos OS issues a warning that the change cannot be made. Failure to modify a protected configuration element does not halt the commit script or the commit process.

Changes that are made to the candidate configuration during the commit operation are not evaluated by the custom rules during that commit operation. However, persistent changes are maintained in the candidate configuration and are evaluated by the custom rules during subsequent commit operations. For more information about how commit scripts change the candidate configuration, see [“Avoiding Potential Conflicts When Using Multiple Commit Scripts”](#) on page 14.

Transient changes are never evaluated by the custom rules in commit scripts, because they are made to the checkout configuration only after the commit scripts have evaluated the candidate configuration and the candidate is copied to become the checkout configuration. To remove a transient change from the configuration, remove, disable, or deactivate the commit script (as discussed in *Controlling Execution of Commit Scripts During Commit Operations*), or comment out the code that generates the transient change.

For more information about differences between persistent and transient changes, see [“Overview of Generating Persistent or Transient Configuration Changes Using Commit Scripts”](#) on page 15.

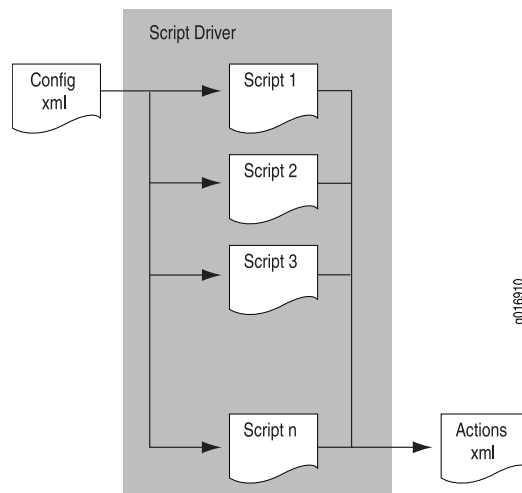
**Related
Documentation**

- [Avoiding Potential Conflicts When Using Multiple Commit Scripts](#) on page 14

Avoiding Potential Conflicts When Using Multiple Commit Scripts

When you use multiple commit scripts, each script evaluates the original candidate configuration file. Changes made by one script are not evaluated by the other scripts. This means that conflicts between scripts might not be resolved when the scripts are first applied to the configuration. The commit scripts are executed in the order they are listed at the **[edit system scripts commit]** hierarchy level, as illustrated in [Figure 4](#) on page 15.

Figure 4: Configuration Evaluation by Multiple Commit Scripts



As an example of a conflict between commit scripts, suppose that commit script **A.xsl** is created to ensure that the device uses the domain name server with IP address **192.168.0.255**. Later, the DNS server's address is changed to **192.168.255.255** and a second script, **B.xsl**, is added to check that the device uses the DNS server with that address. However, script **A.xsl** is not removed or disabled.

Because each commit script evaluates the original candidate configuration, the final result of executing both scripts **A.xsl** and **B.xsl** depends on which DNS server address is configured in the original candidate configuration. If the now outdated address of **192.168.0.255** is configured, script **B.xsl** changes it to **192.168.255.255**. However, if the correct address of **192.168.255.255** is configured, script **A.xsl** changes it to the incorrect value **192.168.0.255**.

As another example of a potential conflict between commit scripts, suppose that a commit script protects a hierarchy using the **protect** attribute. If a second commit script attempts to modify or delete the hierarchy or the statements within the hierarchy, Junos OS issues a warning during the commit process and prevents the configuration change.

Exercise care to ensure that you do not introduce conflicts between scripts like those described in the examples. As a method of checking for conflicts with persistent changes, you can issue two separate **commit** commands.

Related Documentation

- [How Commit Scripts Work on page 9](#)

Overview of Generating Persistent or Transient Configuration Changes

Junos OS commit scripts enforce custom configuration rules. When a candidate configuration includes statements that you have decided must not be included in your configuration, or when the candidate configuration omits statements that you have

decided are required, commit scripts can automatically change the configuration and thereby correct the problem.

- [Differences Between Persistent and Transient Changes on page 16](#)
- [Interaction of Configuration Changes and Configuration Groups on page 19](#)
- [Tag Elements and Templates for Generating Changes on page 19](#)

Differences Between Persistent and Transient Changes

Configuration changes made by commit scripts can be *persistent* or *transient*.

A persistent change remains in the candidate configuration and affects routing operations until you explicitly delete it, even if you subsequently remove or disable the commit script that generated the change and reissue the **commit** command. In other words, removing the commit script does not cause a persistent change to be removed from the configuration.

A transient change, in contrast, is made in the *checkout configuration* but not in the candidate configuration. The checkout configuration is the configuration database that is inspected for standard Junos OS syntax just before it is copied to become the active configuration on the device. If you subsequently remove or disable the commit script that made the change and reissue the **commit** command, the change is no longer made to the checkout configuration and so does not affect the active configuration. In other words, removing the commit script effectively removes a transient change from the configuration.

A common use for transient changes is to eliminate the need to repeatedly configure and display well-known policies, thus allowing these policies to be enforced implicitly. For example, if MPLS must be enabled on every interface with an International Organization for Standardization (ISO) protocol enabled, the change can be transient, so that the repetitive or redundant configuration data need not be carried or displayed in the candidate configuration. Furthermore, transient changes allow you to write script instructions that apply the change only if a set of conditions is met.

Persistent and transient changes are loaded into the configuration in the same manner that the **load replace** configuration mode command loads an incoming configuration. When generating a persistent or transient change, adding the **replace="replace"** attribute to a configuration element produces the same behavior as a **replace:** tag in a **load replace** operation.

By default, Junos OS merges the incoming configuration and the candidate configuration. New statements and hierarchies are added, and conflicting statements are overridden. When generating a persistent or transient change, if you add the **replace="replace"** attribute to a configuration element, Junos OS replaces the existing configuration element with the incoming configuration element. If the **replace="replace"** attribute is added to a configuration element, but there is no existing element of the same name in the current configuration, the incoming configuration element is added into the configuration. Elements that do not have the **replace** attribute are merged into the configuration.

Persistent and transient changes are loaded before the standard Junos validation checks are performed. This means any configuration changes introduced by a commit script are

validated for correct syntax. If the syntax is correct, the new configuration becomes the active, operational device configuration.

Protected elements in the configuration hierarchy cannot be modified or deleted by either a persistent or a transient change. If a commit script attempts to modify or delete a protected statement or hierarchy, Junos OS issues a warning that the change cannot be made, and proceeds with the commit.

Persistent and transient changes have several important differences, as described in [Table 4 on page 17](#).

Table 4: Differences Between Persistent and Transient Changes

| Persistent Changes | Transient Changes |
|--|---|
| <p>A persistent change is represented in a commit script by the <change> tag.</p> <p>Another way to represent a persistent change is with the content parameter inside a call to the jcs:emit-change template.</p> <p>The jcs:emit-change template is a helper template contained in the junos.xsl import file.</p> | <p>A transient change is represented in a commit script by the <transient-change> tag.</p> <p>Another way to represent a transient change is to use the content parameter and the tag transient parameter inside a call to the jcs:emit-change template.</p> |
| <p>You can use persistent changes to perform any Junos XML protocol operation, such as activate, deactivate, delete, insert (reorder), comment (annotate), and replace sections of the configuration.</p> | <p>Like persistent changes, you can use transient changes to perform any Junos XML protocol operation. However, some Junos XML protocol operations do not make sense to use with transient changes, such as generating comments and inactive settings.</p> |
| <p>Persistent changes are always loaded during the commit process if no errors are generated by any commit scripts or by the standard Junos OS validity check.</p> | <p>For transient changes to be loaded, you must include the allow-transients statement at the [edit system scripts commit] hierarchy level. If you enable a commit script that generates transient changes and you do not include the allow-transients statement in the configuration, the CLI generates an error message and the commit operation fails.</p> <p>Like persistent changes, transient changes must pass the standard Junos OS validity check.</p> <p>You cannot use a commit script to generate the allow-transients statement at the [edit system scripts commit] hierarchy level. Rather, you must include this statement directly by using the CLI.</p> |

Table 4: Differences Between Persistent and Transient Changes (*continued*)

| Persistent Changes | Transient Changes |
|---|---|
| <p>Persistent changes work like the load replace configuration mode command, and the change is added to the candidate configuration.</p> <p>When generating a persistent change, if you add the replace="replace" attribute to a configuration element, Junos OS replaces the existing element in the candidate configuration with the incoming configuration element. If there is no existing element of the same name in the candidate configuration, the incoming configuration element is added into the configuration. Elements that do not have the replace attribute are merged into the configuration.</p> | <p>Transient changes work like the load replace configuration mode command, and the change is added to the checkout configuration.</p> <p>When generating a transient change, if you add the replace="replace" attribute to a configuration element, Junos OS replaces the existing element in the checkout configuration with the incoming configuration element. If there is no existing element of the same name in the checkout configuration, the incoming configuration element is added into the configuration. Elements that do not have the replace attribute are merged into the configuration.</p> <p>Transient changes are not copied to the candidate configuration. For this reason, transient changes are not saved in the configuration if the associated commit script is deleted or deactivated.</p> |
| <p>After a persistent change is committed, the software treats it like a change you make by directly editing and committing the candidate configuration.</p> <p>After the persistent changes are copied to the candidate configuration, they are copied to the checkout configuration. If the changes pass the standard Junos OS validity checks, the changes are propagated to the switch, router, or security device components.</p> | <p>Each time a transient change is committed, the software updates the checkout configuration database. After the transient changes pass the standard Junos OS validity checks, the changes are propagated to the device components.</p> |
| <p>After committing a script that causes a persistent change to be generated, you can view the persistent change by issuing the show configuration mode command:</p> <pre>user@host# show</pre> <p>This command displays persistent changes only, not transient changes.</p> | <p>After committing a script that causes a transient change to be generated, you can view the transient change by issuing the show display commit-scripts configuration mode command:</p> <pre>user@host# show display commit-scripts</pre> <p>This command displays both persistent and transient changes.</p> |
| <p>Persistent changes must conform to your custom configuration design rules as dictated by commit scripts.</p> <p>This does not become apparent until after a second commit operation because persistent changes are not evaluated by commit script rules on the current commit operation. The subsequent commit operation fails if the persistent changes do not conform to the rules imposed by the commit scripts configured during the first commit operation.</p> | <p>Transient changes are never tested by and do not need to conform to your custom rules. This is caused by the order of operations in the Junos OS commit model, which is explained in detail in “Commit Scripts and the Junos OS Commit Model” on page 12.</p> |
| <p>A persistent change remains in the configuration even if you delete, disable, or deactivate the commit script instructions that generated the change.</p> | <p>If you delete, disable, or deactivate the commit script instructions that generate a transient change, the change is removed from the configuration after the next commit operation. In short, if the associated instructions or the entire commit script is removed, the transient change is also removed.</p> |

Table 4: Differences Between Persistent and Transient Changes (*continued*)

| Persistent Changes | Transient Changes |
|--|--|
| As with direct CLI configuration, you can remove a persistent change by rolling back to a previous configuration that did not include the change and issuing the commit command. However, if you do not disable or deactivate the associated commit script, and the problem that originally caused the change to be generated still exists, the change is automatically regenerated when you issue another commit command. | You cannot remove a transient change by rolling back to a previous configuration. |
| You can alter persistent changes directly by editing the configuration using the CLI. | <p>You cannot directly alter or delete a transient change by using the Junos OS CLI, because the change is not in the candidate configuration.</p> <p>To alter the contents of a transient change, you must alter the statements in the commit script that generates the transient change.</p> |

Interaction of Configuration Changes and Configuration Groups

Any configuration change you can make by directly editing the configuration using the Junos OS command-line interface (CLI) can also be generated by a commit script as a persistent or transient change. This includes values specified at a specific hierarchy level or in configuration groups. As with direct CLI configuration, values specified in the *target* override values inherited from a configuration group. The target is the statement to which you apply a configuration group by including the **apply-groups** statement.

If you define persistent or transient changes as belonging to a configuration group, the configuration groups are applied in the order you specify in the **apply-groups** statements, which you can include at any hierarchy level except the top level. You can also disable inheritance of a configuration group by including the **apply-groups-except** statement at any hierarchy level except the top level.



CAUTION: Each commit script inspects the postinheritance view of the configuration. If a candidate configuration contains a configuration group, be careful when using a commit script to change the related target configuration, because doing so might alter the intended inheritance from the configuration group.

Also be careful when using a commit script to change a configuration group, because the configuration group might be generated by an application that performs a load replace operation on the group during each commit operation.

For more information about configuration groups, see the *CLI User Guide*.

Tag Elements and Templates for Generating Changes

To generate changes, you can use the **jcs:emit-change** template, which implicitly includes **<change>** and **<transient-change>** XML elements; or you can explicitly include **<change>**

and **<transient-change>** XML elements. Using the **jcs:emit-change** template allows you to set the hierarchical context of the change once rather than multiple times.

The **<change>** and **<transient-change>** elements are similar to the **<load-configuration>** operation defined by the Junos XML management protocol. The possible contents of the **<change>** and **<transient-change>** elements are the same as the contents of the **<configuration>** tag element used in the Junos XML protocol operation **<load-configuration>**. For complete details about the **<load-configuration>** element, see the *Junos XML Management Protocol Developer Guide*.

Required Boilerplate for Commit Scripts

When you write commit scripts, you use Extensible Stylesheet Language Transformations (XSLT) or Stylesheet Language Alternative Syntax (SLAX) tools provided with Junos OS. These tools include basic boilerplate that you must include in all commit scripts, optional extension functions that accomplish scripting tasks more easily, and named templates that make commit scripts easier to read and write, which you import from a file called **junos.xml**. For more information about the extension functions and templates, see *Understanding Extension Functions in the jcs and slax Namespaces* and *Named Templates in the jcs Namespace Overview*.

Commit scripts are based on Junos XML and Junos XML protocol tag elements. Like all XML elements, angle brackets enclose the name of a Junos XML or Junos XML protocol tag element in its opening and closing tags. This is an XML convention, and the brackets are a required part of the complete tag element name. They are not to be confused with the angle brackets used in the documentation to indicate optional parts of Junos OS CLI command strings.

You must include either XSLT or SLAX boilerplate as the starting point for all commit scripts that you create. The XSLT boilerplate follows:

XSLT Boilerplate for Commit Scripts

```
1 <?xml version="1.0" standalone="yes"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:junos="http://xml.juniper.net/junos/*/junos"
5   xmlns:xnm="http://xml.juniper.net/xnm/1.1/xnm"
6   xmlns:jcs="http://xml.juniper.net/junos/commit-scripts/1.0">
7   <xsl:import href="../../../import/junos.xml"/>

8   <xsl:template match="configuration">
9     <!-- ... Insert your code here ... -->
10  </xsl:template>
11 </xsl:stylesheet>
```

Line 1 is the Extensible Markup Language (XML) processing instruction (PI). This PI specifies that the code is written in XML using version 1.0. The XML PI, if present, must be the first noncomment token in the script file.

```
1 <?xml version="1.0"?>
```

Lines 2 through 6 set the style sheet element and the associated namespaces. Line 2 sets the style sheet version as 1.0. Lines 3 through 6 list all the namespace mappings commonly used in commit scripts. Not all of these prefixes are used in this example, but it is not an error to list namespace mappings that are not referenced. Listing all namespace mappings prevents errors if the mappings are used in later versions of the script.

```

2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:junos="http://xml.juniper.net/junos/*/junos"
5   xmlns:xnm="http://xml.juniper.net/xnm/1.1/xnm"
6   xmlns:jcs="http://xml.juniper.net/junos/commit-scripts/1.0">

```

Line 7 is an XSLT import statement. It loads the templates and variables from the file referenced as `../import/junos.xsl`, which ships as part of the Junos OS. The `junos.xsl` file contains a set of named templates you can call in your scripts. These named templates are discussed in *Named Templates in the jcs Namespace Overview* and *Named Templates in the jcs Namespace Summary*.

```

7 <xsl:import href="../import/junos.xsl"/>

```

Line 8 defines a template that matches the `<configuration>` element, which is the node selected by the `<xsl:template match="/">` template, contained in the `junos.xsl` import file. The `<xsl:template match="configuration">` element allows you to exclude the `/configuration/` root element from all XML Path Language (XPath) expressions in the script and begin XPath expressions with the top Junos OS hierarchy level. For more information, see *XPath Overview*.

```

8 <xsl:template match="configuration">

```

Add your code between Lines 8 and 9.

Line 9 closes the template.

```

9 </xsl:template>

```

Line 10 closes the style sheet and the commit script.

```

10 </xsl:stylesheet>

```

SLAX Boilerplate for Commit Scripts

The corresponding SLAX boilerplate is as follows:

```

version 1.0;
ns junos = "http://xml.juniper.net/junos/*/junos";
ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";
ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";
import "../import/junos.xsl";

match configuration {
/*
* Insert your code here
*/
}

```

How Op Scripts Work

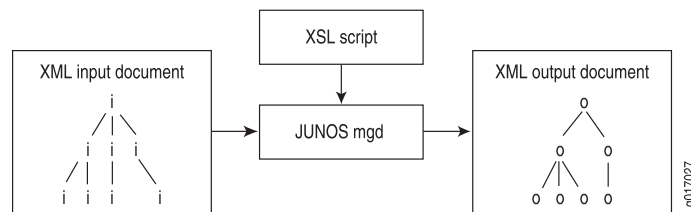
Op scripts execute Junos OS operational commands and inspect the resulting output. After inspection, op scripts can automatically correct errors within the device running Junos OS based on this output.

You add op scripts to device operations by listing the filenames of one or more op script files within the **[edit system scripts op]** hierarchy level. These files must be added to the appropriate op script file directory. For more information about op script file directories, see *Storing Scripts in Flash Memory*. Once added to the device, op scripts are invoked from the command line, using the **op filename** command.

You can use op scripts to generate changes to the device configuration by including the **<load-configuration>** tag element. Because the changes are loaded before the standard validation checks are performed, they are validated for correct syntax, just like statements already present in the configuration before the script is applied. If the syntax is correct, the configuration is activated and becomes the active, operational device configuration.

Figure 5 on page 22 shows a high-level view of the flow of op script input and output.

Figure 5: Op Script Input and Output



Required Boilerplate for Op Scripts

When you write operation (op) scripts, you use Extensible Stylesheet Language Transformations (XSLT) or Stylesheet Language Alternative Syntax (SLAX) tools provided with Junos OS. These tools include basic boilerplate that you must include in all op scripts, optional extension functions that accomplish scripting tasks more easily, and named templates that make scripts easier to read and write, which you import from a file called **junos.xsl**. For more information about the extension functions and templates, see *Understanding Extension Functions in the jcs and slax Namespaces* and *Named Templates in the jcs Namespace Overview*.

Op scripts are based on Junos XML and Junos XML protocol tag elements. Like all XML elements, angle brackets enclose the name of a Junos XML or Junos XML protocol tag element in its opening and closing tags. This is an XML convention, and the brackets are a required part of the complete tag element name. They are not to be confused with the angle brackets used in the documentation to indicate optional parts of Junos OS CLI command strings.

You must include either XSLT or SLAX boilerplate as the starting point for all op scripts that you create. The XSLT boilerplate follows:

XSLT Boilerplate for Op Scripts

```

1 <?xml version="1.0" standalone="yes"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:junos="http://xml.juniper.net/junos/*/junos"
5   xmlns:xnm="http://xml.juniper.net/xnm/1.1/xnm"
6   xmlns:jcs="http://xml.juniper.net/junos/commit-scripts/1.0">
7   <xsl:import href="../import/junos.xsl"/>

```

```

8  <xsl:template match="/">
9    <op-script-results>
      <!-- ... insert your code here ... -->
10   </op-script-results>
11  </xsl:template>
      <!-- ... insert additional template definitions here ... -->
12 </xsl:stylesheet>

```

Line 1 is the Extensible Markup Language (XML) processing instruction (PI), which marks this file as XML and specifies the version of XML as 1.0. The XML PI, if present, must be the first non-comment token in the script file.

```
1  <?xml version="1.0"?>
```

Line 2 opens the style sheet and specifies the XSLT version as 1.0.

```
2  <xsl:stylesheet version="1.0"
```

Lines 3 through 6 list all the namespace mappings commonly used in operation scripts. Not all of these prefixes are used in this example, but it is not an error to list namespace mappings that are not referenced. Listing all namespace mappings prevents errors if the mappings are used in later versions of the script.

```

3  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4  xmlns:junos="http://xml.juniper.net/junos/*/junos"
5  xmlns:xnm="http://xml.juniper.net/xnm/1.1/xnm"
6  xmlns:jcs="http://xml.juniper.net/junos/commit-scripts/1.0">

```

Line 7 is an XSLT import statement. It loads the templates and variables from the file referenced as `../import/junos.xml`, which ships as part of Junos OS (in the file `/usr/libdata/cscript/import/junos.xml`). The `junos.xml` file contains a set of named templates you can call in your scripts. These named templates are discussed in *Named Templates in the jcs Namespace Overview* and *Named Templates in the jcs Namespace Summary*.

```
7  <xsl:import href="../import/junos.xml"/>
```

Line 8 defines a template that matches the `</>` element. The `<xsl:template match="/">` element is the root element and represents the top level of the XML hierarchy. All XML Path Language (XPath) expressions in the script must start at the top level. This allows the script to access all possible Junos XML and Junos XML protocol remote procedure calls (RPCs). For more information, see *XPath Overview*.

```
8  <xsl:template match="/">
```

After the `<xsl:template match="/">` tag element, the `<op-script-results>` and `</op-script-results>` container tags must be the top-level child tags, as shown in Lines 9 and 10.

```

9    <op-script-results>
      <!-- ... insert your code here ... -->
10   </op-script-results>

```

Line 11 closes the template.

```
11  </xsl:template>
```

Between Line 11 and Line 12, you can define additional XSLT templates that are called from within the `<xsl:template match="/">` template.

Line 12 closes the style sheet and the op script.

```
12 </xsl:stylesheet>
```

SLAX Boilerplate for Op Scripts

The corresponding SLAX boilerplate is as follows:

```
version 1.0;
```

```
ns junos = "http://xml.juniper.net/junos/*/junos";  
ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";  
ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";  
import "../import/junos.xsl";
```

```
match / {  
  <op-script-results> {  
    /*  
      * Insert your code here  
    */  
  }  
}
```


CHAPTER 3

Junos Space

- [Understanding Junos Space Support on page 25](#)

Understanding Junos Space Support

The Juniper Networks Junos Space application, running on a JA1500 appliance or a Junos Space Virtual Appliance, is a comprehensive platform for building and deploying applications for collaboration, productivity, and network infrastructure and operations management. Junos Space provides a runtime environment implemented as a fabric of virtual and physical appliances.

The Junos Space Network Management Platform software comprises various applications for network management and configuration, including:

- Junos Space Administration—Provides management of Junos Space fabric, databases, licenses, applications, authentication servers, tags, permission labels, DMI schemas, and troubleshooting.
- Network Director—Provides unified management of supported Juniper Networks devices in your network. By providing full network life cycle management, Network Director simplifies the discovery, configuration, visualization, monitoring, and administration of large networks.
- Service Automation—Provides an end-to-end solution designed to streamline operations and enable proactive network management for Junos OS devices. The solution consists of Advanced Insight Scripts (AI-Scripts), Junos Space Service Now and Service Insight applications, and Juniper Support Systems (JSS).



NOTE: Do not install Junos Space and AI-Scripts on the control plane network EX4200 switches or EX4200 Virtual Chassis in a QFX3000 QFabric system

Before you can use Junos Space Network Director to manage the QFX Series device, you must ensure that the configuration on the device meets the requirements for all managed devices. For example:

- The device configuration has a static management IP address that is reachable from the Junos Space server.
- There is a user with full administrative privileges for Junos Space administration.
- SNMP is enabled (only if you plan on using SNMP as part of the device discovery).
- In Junos Space, set up a default device management interface (DMI) schema for the QFX Series device.

For more information about Network Director requirements, see the *Network Director Quick Start Guide* at:

http://www.juniper.net/techpubs/en_US/network-director1.5/information-products/pathway-pages/index.html

For more information about Junos Space, go to:

http://www.juniper.net/techpubs/en_US/release-independent/junos-space/index.html

**Related
Documentation**

- [Configuring SNMP on page 113](#)
- [Configuring SSH Service for Remote Access to the Router or Switch on page 160](#)

PART 2

Configuration

- [Configuring Network Analytics on page 29](#)
- [Configuring sFlow Technology on page 65](#)
- [Configuring SNMP on page 73](#)
- [Configuring System Logging on page 131](#)
- [Configuration Tasks for Network Management on page 159](#)

CHAPTER 4

Configuring Network Analytics

- [Network Analytics Overview on page 29](#)
- [Understanding Network Analytics Configuration and Status on page 36](#)
- [Understanding Network Analytics Streaming Data on page 38](#)
- [Understanding Enhanced Network Analytics Streaming Data on page 40](#)
- [Understanding Enhanced Analytics Local File Output on page 45](#)
- [Prototype File for the Google Protocol Buffer Stream Format on page 47](#)
- [Example: Configuring Enhanced Network Analytics Features on page 48](#)
- [Configuring Queue Monitoring on page 58](#)
- [Configuring Traffic Monitoring on page 60](#)
- [Configuring a Local File for Network Analytics Data on page 61](#)
- [Configuring a Remote Collector for Streaming Analytics Data on page 62](#)

Network Analytics Overview

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. Network administrators can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed. The analytics manager (analyticsm) in the Packet Forwarding Engine collects traffic and queue statistics, and the analytics daemon (analyticsd) in the Routing Engine analyzes the data and generates reports. You can enable network analytics by configuring microburst monitoring and high-frequency traffic statistics monitoring.



NOTE: In Junos OS Release 13.2X51-D15, the network analytics feature was enhanced, and extensive changes were made to the CLI statements and hierarchies. If you upgrade to Junos OS Release 13.2X51-D15 or later from a release prior to 13.2X51-D15, network analytics configurations committed in previous releases will appear on your device, but the feature is disabled. To enable this feature, you must reconfigure it using the new CLI statements and hierarchies.

For more information, see:

- [Analytics Feature Overview on page 30](#)
- [Network Analytics Enhancements Overview on page 31](#)
- [Summary of CLI Changes on page 32](#)

Analytics Feature Overview

You enable network analytics by configuring queue (microburst) monitoring and high-frequency traffic statistics monitoring. You use microburst monitoring to look at traffic queue conditions in the network. A microburst occurrence indicates to the Packet Forwarding Engine that a user-specified queue depth or latency threshold is reached. The queue depth is the buffer (in bytes) containing the data, and latency is the time (in nanoseconds or microseconds) the data stays in the queue.

You can configure queue monitoring based on either queue depth or latency (but not both), and configure the frequency (polling interval) at which the Packet Forwarding Engine checks for microbursts and sends the data to the Routing Engine for processing. You may configure queue monitoring globally for all physical interfaces on the system, or for a specific interface on the switch. However, the specified queue monitoring interval applies either to all interfaces, or none; you cannot configure the interval for each interface.

You use high-frequency traffic statistics monitoring to collect traffic statistics at specified polling intervals. Similar to the queue monitoring interval, the traffic monitoring interval applies either to all interfaces, or none; you cannot configure the interval for each interface.

Both traffic and queue monitoring are disabled by default. You must configure each type of monitoring using the CLI. In each case, the configuration for an interface always takes precedence over the global configuration.



NOTE: You can configure traffic and queue monitoring for physical interfaces only; logical interfaces and Virtual Chassis port (VCP) interfaces are not supported.

The `analyticsd` daemon in the Routing Engine generates local log files containing queue and traffic statistics records. You can specify the log filename and size, and the number of log files. If you do not configure a filename, the data is not saved.

You can display the local log file or specify a server to receive the streaming data containing the queue and traffic statistics.

For each port, information for the last 10 records of traffic statistics and 100 records of queue statistics is cached. You may view this information by using the **show analytics** commands.

To store traceoptions data, you configure the **traceoptions** statement at the **[edit services analytics]** hierarchy level.

Network Analytics Enhancements Overview

Beginning in Junos OS Release 13.2X51-D15, the network analytics feature provides the following enhancements:

- **Resources**—Consist of interfaces and system. The interfaces resource allows you to configure an interface name and an associated resource profile name for each interface. With the system resource, you can configure the polling intervals for queue monitoring and traffic monitoring, and an associated resource profile for the system.
- **Resource profile**—A template that contains the configurations for queue and traffic monitoring, such as depth threshold and latency threshold values, and whether each type of monitoring is enabled or disabled. Once a resource profile is configured, you apply it to a system or interfaces resource.
- **Collector**—A server for collecting queue and traffic monitoring statistics, and can be a local or remote server. You can configure a local server to store monitoring statistics in a log file, or a remote server to receive streamed statistics data.
- **Export profile**—You must configure an export profile if you wish to send streaming data to a remote collector. In the export profile, you define the category of streamed data (system-wide or interface-specific) to determine stream type the collector will receive. You can specify both system and interface stream categories. System data includes system information and status of queue and traffic monitoring. Interface-specific data includes interface information, queue and traffic statistics, and link, queue, and traffic status.
- **Google Protocol Buffer (GBP) stream format**—A new streaming format for monitoring statistics data that is sent to a remote collector in a single AnRecord message. This stream format provides nine types of information, including:
 - **System information**—General system information, including boot time, model information, serial number, number of ports, and so on.
 - **System queue status**—Queue status for the system in general.
 - **System traffic status**—Traffic status for the system in general.
 - **Interface information**—Includes SNMP index, slot, port, and other information.
 - **Queue statistics for interfaces**—Queue statistics for specific interfaces.
 - **Traffic statistics for interfaces**—Traffic statistics for specific interfaces.
 - **Link status for interfaces**—Includes link speed, state, and so on.
 - **Queue status for interfaces**—Queue status for specific interfaces.
 - **Traffic status for interfaces**—Traffic status for specific interfaces.
- **The `analytics.proto` file**—Provides a template for the GBP stream format. This file can be used for writing your analytics server application. To download the file, go to:
http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/proto-files/analytics-proto.txt
- **Use of threshold values**—The Analytics Manager (analyticsm) will generate a queue statistics record when the lower queue depth or latency threshold value is exceeded.

- User Datagram Protocol (UDP)—Additional transport protocol you can configure, in addition to Transmission Control Protocol (TCP), for the remote streaming server port.
- Single file for local logging—Replaces the separate log files for queue and traffic statistics.
- Change in latency measurement—Configuration and reporting of latency values have changed from microseconds to nanoseconds.
- Change in reporting of the collection time in UTC format—Statistics collection time is reported in microseconds instead of milliseconds.
- New operational mode command **show analytics collector**—Replaces the **show analytics streaming-server** command.
- Changes in command output format—Include the following changes:
 - Addition of unicast, multicast, and broadcast packet counters in queue and traffic statistics.
 - Reversal of the sequence of statistics information in the output. The most recent record is displayed at the beginning, and the oldest record at the end of the output.
 - Removal of traffic or queue monitoring status information from the global portion of the **show analytics configuration** and **show analytics status** command output if there is no global configuration.
 - Addition of **n/a** to the interface-specific portion of the **show analytics configuration** and **show analytics status** command output if a parameter is not configured (for example, depth threshold or latency threshold).

Summary of CLI Changes

Beginning in Junos OS Release 13.2X51-D15, enhancements to the network analytics feature result in changes in the CLI when you configure the feature. See [Table 5 on page 32](#) for a summary of CLI changes.

Table 5: Network Analytics CLI Changes

| Task | CLI for Junos OS Release 13.2X50-D15 and 13.2X51-D10 | CLI for Junos OS Release 13.2X51-D15 and later |
|--|--|---|
| Configuring global queue and traffic monitoring polling interval | <pre>[edit services analytics] traffic-statistics { interval <i>interval</i>; } queue-statistics { interval <i>interval</i>; }</pre> | <pre>[edit services analytics] resource { system { polling-interval { queue-monitoring <i>interval</i>; traffic-monitoring <i>interval</i>; } } }</pre> |

Table 5: Network Analytics CLI Changes (*continued*)

| Task | CLI for Junos OS Release 13.2X50-D15 and 13.2X51-D10 | CLI for Junos OS Release 13.2X51-D15 and later |
|--|--|--|
| Configuring local files for traffic and queue statistics reporting | <pre>[edit services analytics] traffic-statistics { file <i>filename</i>; size <i>size</i>; files <i>number</i>; } queue-statistics { file <i>filename</i>; size <i>size</i>; files <i>number</i>; }</pre> | <pre>[edit services analytics] collector { local { file <i>filename</i> { files <i>number</i>; size <i>size</i>; } } }</pre> |
| Enabling queue statistics and traffic monitoring, and specifying the depth threshold for all interfaces (globally) | <pre>[edit services analytics] interfaces { all { queue-statistics; traffic-statistics; depth-threshold { high <i>number</i>; low <i>number</i>; } } }</pre> | <p>Requires defining a resource profile and applying it to the system:</p> <ol style="list-style-type: none"> To define a resource profile: <pre>[edit services analytics] resource-profiles { <i>profile-name</i> { queue-monitoring; traffic-monitoring; depth-threshold { high <i>number</i>; low <i>number</i>; } } }</pre> To apply a profile to the system: <pre>[edit services analytics] resource { system { resource-profile <i>profile-name</i>; } }</pre> |

Table 5: Network Analytics CLI Changes (*continued*)

| Task | CLI for Junos OS Release 13.2X50-D15 and 13.2X51-D10 | CLI for Junos OS Release 13.2X51-D15 and later |
|---|--|---|
| Enabling queue statistics and traffic monitoring, and specifying the latency threshold for one interface | <pre>[edit services analytics] interfaces { interface { queue-statistics; traffic-statistics; latency-threshold high <i>number</i>; low <i>number</i>; } }</pre> | <p>Requires defining a resource profile and applying it to the interface:</p> <ol style="list-style-type: none"> To define a resource profile: <pre>[edit services analytics] resource-profiles { profile-name { queue-monitoring; traffic-monitoring; latency-threshold { high <i>number</i>; low <i>number</i>; } } }</pre> To apply a profile to the interface: <pre>[edit services analytics] resource { interfaces { interface-name { resource-profile <i>profile-name</i>; } } }</pre> |
| <p>Configuring the streaming data format (JSON, CSV, or TSV) to send to a remote server</p> <p>NOTE: Junos OS Release 13.2X51-D15 added support for the GPB stream format and configuration of the transport protocols (TCP or UDP).</p> | <pre>[edit services analytics] streaming-servers { address <i>ip-address</i> { port <i>number</i> { stream-format <i>format</i>; } } }</pre> | <p>Requires defining the stream format in an export profile and applying the profile to the collector.</p> <ol style="list-style-type: none"> To configure the stream format: <pre>[edit services analytics] export-profiles { profile-name { stream-format <i>format</i>; } }</pre> To apply an export profile to the collector: <pre>[edit services analytics] collector { address <i>ip-address</i> { port <i>number</i> { transport <i>protocol</i> { export-profile <i>profile-name</i>; } } } }</pre> |

Table 5: Network Analytics CLI Changes (*continued*)

| Task | CLI for Junos OS Release 13.2X50-D15 and 13.2X51-D10 | CLI for Junos OS Release 13.2X51-D15 and later |
|--|---|--|
| Configuring the streaming message types (queue or traffic statistics) to send to a remote server | <pre> [edit services analytics] streaming-servers { address <i>ip-address</i> { port <i>number</i> { stream-type <i>type</i>; stream-type <i>type</i>; } } } </pre> | <p>Requires defining an export profile and applying it to the collector:</p> <ol style="list-style-type: none"> To define an export profile: <pre> [edit services analytics] export-profiles { <i>profile-name</i> { interface { information; statistics { queue; traffic; } status { link; queue; traffic; } } } system { information; status { queue; traffic; } } } </pre> To apply an export profile to the collector: <pre> [edit services analytics] collector { address <i>ip-address</i> { port <i>number</i> { export-profile <i>profile-name</i>; } } } </pre> |

Table 5: Network Analytics CLI Changes (*continued*)

| Task | CLI for Junos OS Release 13.2X50-D15 and 13.2X51-D10 | CLI for Junos OS Release 13.2X51-D15 and later |
|---|--|---|
| Configuring the transport protocol for sending streaming data to an external server | No configuration is available. Only the TCP protocol is supported. | Configuration is available. Both TCP and UDP protocols are supported, and can be configured for the same port. [edit services analytics] collector { address <i>ip-address</i> { port <i>number1</i> { transport tcp; transport udp; } port <i>number2</i> { transport udp; } } } |
| Show information about remote streaming server or collector | Issue the show analytics streaming-server command. | Issue the show analytics collector command. |

Related Documentation

- [analytics](#)

Understanding Network Analytics Configuration and Status

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. You can enable network analytics by configuring traffic and queue statistics monitoring.



NOTE: This topic describes the configuration and status output from Junos OS Release 13.2X50-D15 and 13.2X51-D10 only.

If you had enabled traffic or queue monitoring, you can issue the **show analytics configuration** and **show analytics status** commands to view the global interface configuration and status and that of specific interfaces. The output that is displayed depends on your configuration at the global interface and specific interface levels. For example:

- A global interface configuration (for all interfaces) to disable monitoring supersedes the configuration to enable it on an interface.
- The interface configuration to enable or disable monitoring supersedes the global interface configuration, unless monitoring had been disabled globally for all interfaces.
- If there is no configuration, whether for all interfaces or a specific interface, monitoring is disabled by default (see [Table 6 on page 37](#)).

Table 6 on page 37 describes the correlation between the user configuration and the settings that are displayed.

Table 6: Configuration and Status Output in Junos OS Release 13.2X51-D10 and 13.2X50-D15

| User Configuration | Global or System Settings | | Specific Interface Settings | |
|---|---------------------------|----------|-----------------------------|----------|
| | Configuration | Status | Configuration | Status |
| No global or specific interface configuration. This is the default setting. | Auto | Auto | Auto | Disabled |
| No global interface configuration but the specific interface monitoring is disabled. | Auto | Auto | Disabled | Disabled |
| No global interface configuration but the specific interface monitoring is enabled. | Auto | Auto | Enabled | Enabled |
| Monitoring is disabled globally and there is no interface configuration. | Disabled | Disabled | Auto | Disabled |
| Monitoring is disabled at both the global and specific interface levels. | Disabled | Disabled | Disabled | Disabled |
| Monitoring is disabled at the global interface level but is enabled at the specific interface level. The global interface <i>Disabled</i> setting supersedes the <i>Enabled</i> setting for a specific interface. | Disabled | Disabled | Enabled | Disabled |
| Monitoring is enabled for all interfaces but there is no configuration for the specific interface . | Enabled | Enabled | Auto | Enabled |
| Monitoring is enabled at both the global and specific interface levels. | Enabled | Enabled | Enabled | Enabled |
| Monitoring is enabled for all interfaces but is disabled for the specific interface. | Enabled | Enabled | Disabled | Disabled |

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - *analytics*
 - *queue-statistics*
 - *traffic-statistics*
 - *show analytics configuration*
 - *show analytics status*

Understanding Network Analytics Streaming Data

This topic describes the network analytics queue and traffic statistics that are streamed to remote servers.

You can configure one or more remote servers to receive streamed data containing queue and traffic statistics. The format of the streamed data can be Javascript Object Notation (JSON), Comma-separated Values (CSV), or Tab-separated Values (TSV).



NOTE: The output shown in this topic applies to Junos OS Release 13.2X51-D10 only. The time is displayed in the Unix epoch format (also known as Unix time or POSIX time).

The following examples show the streamed queue statistics data output in different formats.

- JSON format:

```
{"record-type":"queue-stats","time":1383453988263,"router-id":"qfx5100-switch",
"port":"xe-0/0/18","latency":0,"queue-depth":208}
```

- CSV format:

```
q,1383454067604,qfx5100-switch,xe-0/0/18,0,208
```

- TSV format:

```
q      585870192561703872      qfx5100-switch      xe-0/0/18      (null)
208    2
```

[Table 7 on page 38](#) describes the output fields for streamed queue statistics data in the order they appear.

Table 7: Streamed Queue Statistics Data Output Fields

| Field | Description |
|-------------|--|
| record-type | Type of statistics. Displayed as: <ul style="list-style-type: none"> queue-stats (JSON format) q (CSV or TSV format) |
| time | Time (in Unix epoch format) at which the statistics were captured. |
| router-id | ID of the network analytics host device. |
| port | Name of the physical port configured for network analytics. |
| latency | Traffic queue latency in milliseconds. |
| queue depth | Depth of the traffic queue in bytes. |

The following examples show the streamed traffic statistics data output in different formats.

- JSON format:

```
{"record-type":"traffic-stats","time":1383453986763,"router-id":"qfx5100-switch",
"port":"xe-0/0/16","rxpkt":26524223621,"rxpps":8399588,"rxbyte":3395100629632,
"rxbps":423997832,"rxdrop":0,"rxerr":0,"txpkt":795746503,"txpps":0,"txbyte":101855533467,
"txbps":0,"txdrop":0,"txerr":0}
```

- CSV format:

```
t,1383454072924,qfx5100-switch,xe-0/0/19,1274299748,82950,163110341556,85603312,0,0,
27254178291,8300088,3488534810679,600002408,27268587050,3490379142400
```

- TSV format:

```
t      1383454139025    qfx5100-switch  xe-0/0/19      1279874033      82022
163823850036    84801488      0      0      27811618258    8199630
3559887126455    919998736      27827356915    3561901685120
```

[Table 8 on page 39](#) describes the output fields for streamed traffic statistics data in the order they appear.

Table 8: Streamed Traffic Statistics Data Output Fields

| Field | Description |
|-------------|--|
| record-type | Type of statistics. Displayed as: <ul style="list-style-type: none"> • traffic-stats (JSON format) • t (CSV or TSV format) |
| time | Time (in Unix epoch format) at which the statistics were captured. |
| router-id | ID of the network analytics host device. |
| port | Name of the physical port configured for network analytics. |
| rxpkt | Total packets received. |
| rxpps | Total packets received per second. |
| rxbyte | Total bytes received. |
| rxbps | Total bytes received per second. |
| rxdrop | Total incoming packets dropped. |
| rxerr | Total packets with errors. |
| txpkt | Total packets transmitted. |
| txpps | Total packets transmitted per second. |
| txbyte | Total bytes transmitted. |

Table 8: Streamed Traffic Statistics Data Output Fields (*continued*)

| Field | Description |
|--------|--|
| txbps | Total bytes transmitted per second. |
| txdrop | Total transmitted bytes dropped. |
| txerr | Total transmitted packets with errors (dropped). |

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - *show analytics streaming-servers*
 - *streaming-servers*

Understanding Enhanced Network Analytics Streaming Data

Network analytics monitoring data can be streamed to remote servers called collectors. You can configure one or more collectors to receive streamed data containing queue and traffic statistics. This topic describes the streamed data output.



NOTE: This topic applies to Junos OS Release 13.2X51-D15 or later.

Starting in Junos OS Release 13.2X51-D15, network analytics supports the following streaming data formats and output:

- [Google Protocol Buffer \(GPB\) on page 40](#)
- [JavaScript Object Notation \(JSON\) on page 43](#)
- [Comma-separated Values \(CSV\) on page 43](#)
- [Tab-separated Values \(TSV\) on page 43](#)
- [Queue Statistics Output for JSON, CSV, and TSV on page 44](#)
- [Traffic Statistics Output for JSON, CSV, and TSV on page 44](#)

Google Protocol Buffer (GPB)

Support for the Google Protocol Buffer (GPB) streaming format has been added in Junos OS Release 13.2X51-D15. This streaming format provides:

- Support for nine types of messages, based on resource type (system-wide or interface-specific).
- Sends messages in a hierarchical format.
- You can generate other stream format messages (JSON, CSV, TSV) from GPB formatted messages.
- Includes a 8-byte message header. See [Table 9 on page 41](#) for more information.

Table 9 on page 41 describes the GPB stream format message header.

Table 9: GPB Stream Format Message Header Information

| Byte Position | Field |
|---------------|-------------------------|
| 0 to 3 | Length of message |
| 4 | Message version |
| 5 to 7 | Reserved for future use |

The following GPB prototype file (**analytics.proto**) provides details about the streamed data:

```
package analytics;

// Traffic statistics related info
message TrafficStatus {
    optional uint32      status      = 1;
    optional uint32      poll_interval = 2;
}

// Queue statistics related info
message QueueStatus {
    optional uint32      status      = 1;
    optional uint32      poll_interval = 2;
    optional uint64      lt_high     = 3;
    optional uint64      lt_low      = 4;
    optional uint64      dt_high     = 5;
    optional uint64      dt_low      = 6;
}

message LinkStatus {
    optional uint64      speed        = 1;
    optional uint32      duplex       = 2;
    optional uint32      mtu          = 3;
    optional bool        state        = 4;
    optional bool        auto_negotiation = 5;
}

message InterfaceInfo {
    optional uint32      snmp_index   = 1;
    optional uint32      index        = 2;
    optional uint32      slot         = 3;
    optional uint32      port         = 4;
    optional uint32      media_type   = 5;
    optional uint32      capability   = 6;
    optional uint32      porttype     = 7;
}

message InterfaceStatus {
    optional LinkStatus   link        = 1;
    optional QueueStatus  queue_status = 2;
    optional TrafficStatus traffic_status = 3;
}

message QueueStats {
    optional uint64      timestamp    = 1;
```

```

        optional uint64      queue_depth    = 2;
        optional uint64      latency        = 3;
    }

    message TrafficStats {
        optional uint64      timestamp      = 1;
        optional uint64      rxpkt          = 2;
        optional uint64      rxucpkt       = 3;
        optional uint64      rxmcpkt       = 4;
        optional uint64      rxbcpkt       = 5;
        optional uint64      rxpps         = 6;
        optional uint64      rxbyte        = 7;
        optional uint64      rxbps         = 8;
        optional uint64      rxrcerr       = 9;
        optional uint64      rxdropkt      = 10;
        optional uint64      txpkt         = 11;
        optional uint64      txucpkt       = 12;
        optional uint64      txmcpkt       = 13;
        optional uint64      txbcpkt       = 14;
        optional uint64      txpps         = 15;
        optional uint64      txbyte        = 16;
        optional uint64      txbps         = 17;
        optional uint64      txrcerr       = 18;
        optional uint64      txdropkt      = 19;
    }

    message InterfaceStats {
        optional TrafficStats traffic_stats = 1;
        optional QueueStats  queue_stats  = 2;
    }

    //Interface message
    message Interface {
        required string      name          = 1;
        optional bool        deleted       = 2;
        optional InterfaceInfo information = 3;
        optional InterfaceStats stats      = 4;
        optional InterfaceStatus status    = 5;
    }

    message SystemInfo {
        optional uint64      boot_time     = 1;
        optional string      model_info    = 2;
        optional string      serial_no     = 3;
        optional uint32      max_ports     = 4;
        optional string      collector     = 5;
        repeated string      interface_list = 6;
    }

    message SystemStatus {
        optional QueueStatus queue_status = 1;
        optional TrafficStatus traffic_status = 2;
    }

    //System message
    message System {
        required string      name          = 1;
        optional bool        deleted       = 2;
        optional SystemInfo  information = 3;
        optional SystemStatus status      = 4;
    }

```

```

message AnRecord {
    optional uint64      timestamp      = 1;
    optional System      system        = 2;
    repeated Interface   interface     = 3;
}

```

JavaScript Object Notation (JSON)

The JavaScript Object Notation (JSON) streaming format supports the following data:

- Queue statistics data. For example:

```

{"record-type":"queue-stats","time":1383453988263,"router-id":"qfx5100-switch",
"port":"xe-0/0/18","latency":0,"queue-depth":208}

```

See [Table 7 on page 38](#) for more information about queue statistics output fields.

- Traffic statistics. For example:

```

{"record-type":"traffic-stats","time":1383453986763,"router-id":"qfx5100-switch",
"port":"xe-0/0/16","rxpkt":26524223621,"rxpps":8399588,"rxbyte":3395100629632,
"rxbps":423997832,"rxdrop":0,"rxerr":0,"txpkt":795746503,"txpps":0,"txbyte":101855533467,
"txbps":0,"txdrop":0,"txerr":0}

```

See [Table 8 on page 39](#) for more information about traffic statistics output fields.

Comma-separated Values (CSV)

The Comma-separated Values (CSV) streaming format supports the following data:

- Queue statistics. For example:

```

q,1383454067604,qfx5100-switch,xe-0/0/18,0,208

```

See [Table 7 on page 38](#) for more information about queue statistics output fields.

- Traffic statistics. For example:

```

t,1383454072924,qfx5100-switch,xe-0/0/19,1274299748,82950,163110341556,85603312,0,0,
27254178291,8300088,3488534810679,600002408,27268587050,3490379142400

```

See [Table 8 on page 39](#) for more information about traffic statistics output fields.

Tab-separated Values (TSV)

The Tab-separated Values (TSV) streaming format supports the following data:

- Queue statistics. For example:

```

q      585870192561703872      qfx5100-switch      xe-0/0/18      (null)
208      2

```

See [Table 7 on page 38](#) for more information about queue statistics output fields.

- Traffic statistics. For example:

```

t      1383454139025      qfx5100-switch      xe-0/0/19      1279874033      82022
163823850036      84801488      0      0      27811618258      8199630
3559887126455      919998736      27827356915      3561901685120

```

See [Table 8 on page 39](#) for more information about traffic statistics output fields.

Queue Statistics Output for JSON, CSV, and TSV

Table 7 on page 38 describes the output fields for streamed queue statistics data in the order they appear.

Table 10: Streamed Queue Statistics Data Output Fields

| Field | Description |
|-------------|--|
| record-type | Type of statistics. Displayed as: <ul style="list-style-type: none"> • queue-stats (JSON format) • q (CSV or TSV format) |
| time | Time (in Unix epoch format) at which the statistics were captured. |
| router-id | ID of the network analytics host device. |
| port | Name of the physical port configured for network analytics. |
| latency | Traffic queue latency in milliseconds. |
| queue depth | Depth of the traffic queue in bytes. |

Traffic Statistics Output for JSON, CSV, and TSV

Table 8 on page 39 describes the output fields for streamed traffic statistics data in the order they appear.

Table 11: Streamed Traffic Statistics Data Output Fields

| Field | Description |
|-------------|--|
| record-type | Type of statistics. Displayed as: <ul style="list-style-type: none"> • traffic-stats (JSON format) • t (CSV or TSV format) |
| time | Time (in Unix epoch format) at which the statistics were captured. |
| router-id | ID of the network analytics host device. |
| port | Name of the physical port configured for network analytics. |
| rxpkt | Total packets received. |
| rxpps | Total packets received per second. |
| rxbyte | Total bytes received. |
| rxbps | Total bytes received per second. |

Table 11: Streamed Traffic Statistics Data Output Fields (*continued*)

| Field | Description |
|--------|--|
| rxdrop | Total incoming packets dropped. |
| rxerr | Total packets with errors. |
| txpkt | Total packets transmitted. |
| txpps | Total packets transmitted per second. |
| txbyte | Total bytes transmitted. |
| txbps | Total bytes transmitted per second. |
| txdrop | Total transmitted bytes dropped. |
| txerr | Total transmitted packets with errors (dropped). |

**Related
Documentation**

- [Network Analytics Overview on page 29](#)
- [Prototype File for the Google Protocol Buffer Stream Format on page 47](#)
- *address (Analytics Collector)*
- *collector (Analytics)*
- *show analytics collector*

Understanding Enhanced Analytics Local File Output

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. You enable network analytics by configuring queue or traffic statistics monitoring, or both. In addition, you can configure a local file for storing the traffic and queue statistics records.



NOTE: This topic describes the local file output in Junos OS Release 13.2X51-D15 and later. For information about local file output from earlier releases, see the *monitor start (Analytics)* topic.

Beginning in Junos OS Release 13.2X51-D15, the traffic and queue monitoring statistics can be stored locally in a single file. The following example shows the output from the **monitor start** command.

```
root@qfx5100-33> monitor start an
root@qfx5100-33>
*** an ***
q,1393947567698432,qfx5100-33,xe-0/0/19,1098572,1373216
q,1393947568702418,qfx5100-33,xe-0/0/19,1094912,1368640
q,1393947569703415,qfx5100-33,xe-0/0/19,1103065,1378832
```

```
t,1393947569874528,qfx5100-33,xe-0/0/16,12603371884,12603371884,0,0,
8426023,1613231610488,8628248712,0,3,5916761,5916761,0,0,0,757345408,0,0,0
t,1393947569874528,qfx5100-33,xe-0/0/18,12601953614,12601953614,0,0,
8446737,1613050071660,8649421552,0,5,131761619,131761619,0,0,84468,
16865487232,86495888,0,0
t,1393947569874528,qfx5100-33,xe-0/0/19,126009250,126009250,0,0,84469,
16129184128,86496392,0,0,12584980342,12584980342,0,0,8446866,1610877487744,
8649588432,12593703960,0
q,1393947575698402,qfx5100-33,xe-0/0/19,1102233,1377792
q,1393947576701398,qfx5100-33,xe-0/0/19,1107724,1384656
```

See [Table 12 on page 46](#) for queue statistics output, and [Table 13 on page 46](#) for traffic statistics output. The fields in the tables are listed in the order they appear in the output example.

Table 12: Output Fields for Queue Statistics in Local Analytics File

| Field | Description | Example in Output |
|-----------------------|--|-------------------------|
| Record type | Type of statistics (queue or traffic monitoring) | q |
| Time (microseconds) | Unix epoch (or Unix time) in microseconds at which the statistics were captured. | 1393947567698432 |
| Router ID | ID of the network analytics host device. | qfx5100-33 |
| Port | Name of the physical port configured for network analytics. | xe-0/0/19 |
| Latency (nanoseconds) | Traffic queue latency in nanoseconds. | 1098572 |
| Queue depth (bytes) | Depth of the traffic queue in bytes. | 1373216 |

Table 13: Output Fields for Traffic Statistics in Local Analytics File

| Field | Description | Example in Output |
|---------------------|--|-------------------------|
| Record type | Type of statistics (queue or traffic monitoring) | t |
| Time (microseconds) | Unix epoch (or Unix time) in microseconds at which the statistics were captured. | 1393947569874528 |
| Router ID | ID of the network analytics host device. | qfx5100-33 |
| Port | Name of the physical port configured for network analytics. | xe-0/0/16 |
| rxpkt | Total packets received. | 12603371884 |
| rxucpkt | Total unicast packets received. | 12603371884 |
| rxmcpkt | Total multicast packets received. | 0 |
| rxbcpkt | Total broadcast packets received. | 0 |

Table 13: Output Fields for Traffic Statistics in Local Analytics File (*continued*)

| Field | Description | Example in Output |
|-----------|---------------------------------------|-------------------|
| rxpps | Total packets received per second. | 8426023 |
| rxbyte | Total octets received. | 1613231610488 |
| rxbps | Total bytes received per second. | 8628248712 |
| rxdroppkt | Total incoming packets dropped. | 0 |
| rxrcerr | CRC/Align errors received. | 3 |
| txpkt | Total packets transmitted. | 5916761 |
| txucpkt | Total unicast packets transmitted. | 5916761 |
| txmcpkt | Total multicast packets transmitted. | 0 |
| txbcpkt | Total broadcast packets transmitted. | 0 |
| txpps | Total packets transmitted per second. | 0 |
| txbyte | Total octets transmitted. | 757345408 |
| txbps | Bytes per second transmitted. | 0 |
| txdroppkt | Total transmitted packets dropped. | 0 |
| txrcerr | CRC/Align errors transmitted. | 0 |

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - *analytics*

Prototype File for the Google Protocol Buffer Stream Format

The Google Protocol Buffer (GBP) stream format is used for streaming monitoring statistics data to a remote collector in a single AnRecord message.

The **analytics.proto** file provides a template for the GBP stream format. This file can be used for writing your analytics server application.

To download the GBP prototype file, go to:

http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/proto-files/analytics-proto.txt

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - *analytics*

- *export-profiles*

Example: Configuring Enhanced Network Analytics Features

This example shows how to configure the enhanced network analytics feature, including queue and traffic monitoring.

- [Requirements on page 48](#)
- [Overview on page 48](#)
- [Configuration on page 49](#)
- [Verification on page 54](#)

Requirements

This example uses the following hardware and software components:

- A QFX5100 standalone switch
- A external streaming server to collect data
- Junos OS Release 13.2X51-D15 software
- TCP server software (for remote streaming servers)

Before you configure network analytics, be sure you have:

- Junos OS Release 13.2X51-D15 or later software installed and running on the QFX5100 switch.
- (Optional for streaming servers for the JSON, CSV, and TSV formats) TCP or UDP server software set up for processing records separated by a newline character (\n) on the remote streaming server.
- (Optional for streaming servers for the GPB format) TCP or UDP build streaming server using the **analytics.proto** file.
- All other network devices running.

Overview

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. Network administrators can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed.

You enable network analytics by first defining a resource profile template, and then applying the profile to the system (for a global configuration) or to individual interfaces.



NOTE: You can configure queue and traffic monitoring on physical network interfaces only; logical interfaces and Virtual Chassis physical (VCP) interfaces are not supported.

Disabling of the queue or traffic monitoring supersedes the configuration (enabling) of this feature. You disable monitoring by applying a resource profile that includes the `no-queue-monitoring` or `no-traffic-monitoring` configuration statement at the `[edit services analytics resource-profiles]` hierarchy level.

Topology

In this example, the QFX5100 switch is connected to an external server used for streaming statistics data.

Configuration

To configure the network analytics features, perform these tasks:

- [Configuring the Polling Interval for Queue and Traffic Monitoring on page 50](#)
- [Configuring a Local Statistics File on page 50](#)
- [Configuring and Applying a Resource Profile for the System on page 50](#)
- [Configuring and Applying a Resource Profile for an Interface on page 51](#)
- [Configuring an Export Profile and Collector for Streaming Data on page 51](#)

CLI Quick Configuration

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

```
[edit]
set services analytics resource system polling-interval queue-monitoring 1000
set services analytics resource system polling-interval traffic-monitoring 5
set services analytics collector local file an.stats
set services analytics collector local file an files 3
set services analytics collector local file an size 10m
set services analytics resource-profiles sys-rp queue-monitoring
set services analytics resource-profiles sys-rp traffic-monitoring
set services analytics resource-profiles sys-rp depth-threshold high 999999 low 99
set services analytics resource system resource-profile sys-rp
set services analytics resource-profiles if-rp queue-monitoring
set services analytics resource-profiles if-rp traffic-monitoring
set services analytics resource-profiles if-rp latency-threshold high 2300 low 20
set services analytics resource interfaces xe-0/0/16 resource-profile if-rp
set services analytics resource interfaces xe-0/0/18 resource-profile if-rp
set services analytics resource interfaces xe-0/0/19 resource-profile if-rp
set services analytics export-profiles ep stream-format gpb
set services analytics export-profiles ep interface information
set services analytics export-profiles ep interface statistics queue
set services analytics export-profiles ep interface statistics traffic
set services analytics export-profiles ep interface status link
```

```
set services analytics export-profiles ep system information
set services analytics export-profiles ep system status queue
set services analytics export-profiles ep system status traffic
set services analytics collector address 10.94.198.11 port 50001 transport tcp export-profile
ep
set services analytics collector address 10.94.184.25 port 50013 transport udp
export-profile ep
```

Configuring the Polling Interval for Queue and Traffic Monitoring

Step-by-Step Procedure

To configure the polling interval queue and traffic monitoring globally:

1. Configure the queue monitoring polling interval (in milliseconds) for the system:
[edit]
set services analytics resource system polling-interval queue-monitoring 1000
2. Configure the traffic monitoring polling interval (in seconds) for the system:
[edit]
set services analytics resource system polling-interval traffic-monitoring 5

Configuring a Local Statistics File

Step-by-Step Procedure

To configure a file for local statistics collection:

1. Configure the filename:
[edit]
set services analytics collector local file an.stats
2. Configure the number of files:
[edit]
set services analytics collector local file an files 3
3. Configure the file size:
[edit]
set services analytics collector local file an size 10m

Configuring and Applying a Resource Profile for the System

Step-by-Step Procedure

To define a resource profile template for queue and traffic monitoring resources:

1. Configure a resource profile and enable queue monitoring:
[edit]
set services analytics resource-profiles sys-rp queue-monitoring
2. Enable traffic monitoring in the profile:
[edit]
set services analytics resource-profiles sys-rp traffic-monitoring
3. Configure the depth-threshold (high and low values) for queue monitoring in the profile:
[edit]

```
set services analytics resource-profiles sys-rp depth-threshold high 999999 low 99
```

4. Apply the resource profile template to the system resource type for a global configuration:

```
[edit]
set services analytics resource system resource-profile sys-rp
```

Configuring and Applying a Resource Profile for an Interface

Step-by-Step Procedure

You can configure queue and traffic monitoring for one or more specific interfaces. The interface-specific configuration supersedes the global (system) configuration. To define a resource profile template for queue and traffic monitoring resources for an interface:

1. Configure a resource profile and enable queue monitoring:

```
[edit]
set services analytics resource-profiles if-rp queue-monitoring
```

2. Enable traffic monitoring in the profile:

```
[edit]
set services analytics resource-profiles if-rp traffic-monitoring
```

3. Configure the latency-threshold (high and low values) for queue monitoring in the profile:

```
[edit]
set services analytics resource-profiles if-rp latency-threshold high 2300 low 20
```

4. Apply the resource profile template to the interfaces resource type for specific interfaces:

```
[edit]
set services analytics resource interfaces xe-0/0/16 resource-profile if-rp
set services analytics resource interfaces xe-0/0/18 resource-profile if-rp
set services analytics resource interfaces xe-0/0/19 resource-profile if-rp
```

Configuring an Export Profile and Collector for Streaming Data

Step-by-Step Procedure

To configure a collector (streaming server) for receiving monitoring data:

1. Create an export profile and specify the stream format:

```
[edit]
set services analytics export-profiles ep stream-format gpb
```

2. Configure the export profile to include interface information:

```
[edit]
set services analytics export-profiles ep interface information
```

3. Configure the export profile to include interface queue statistics:

```
[edit]
set services analytics export-profiles ep interface statistics queue
```

4. Configure the export profile to include interface traffic statistics:

```
[edit]
set services analytics export-profiles ep interface statistics traffic
```

5. Configure the export profile to include interface status link information:

```
[edit]
set services analytics export-profiles ep interface status link
```

6. Configure the export profile to include system information:

```
[edit]
set services analytics export-profiles ep system information
```

7. Configure the export profile to include system queue status:

```
[edit]
set services analytics export-profiles ep system status queue
```

8. Configure the export profile to include system traffic status:

```
[edit]
set services analytics export-profiles ep system status traffic
```

9. Configure the transport protocol for the collector addresses and apply an export profile:

```
[edit]
set services analytics collector address 10.94.198.11 port 50001 transport tcp
export-profile ep
set services analytics collector address 10.94.184.25 port 50013 transport udp
export-profile ep
```



NOTE: If you configure the `tcp` or `udp` option for the JSON, CSV, and TSV formats, you must also set up the TCP or UDP client software on the remote collector to process records that are separated by the newline character (`\n`) on the remote server.

If you configure the `tcp` or `udp` option for the GPB format, you must also set up the TCP or UDP build streaming server using the `analytics.proto` file.

Results Display the results of the configuration:

```
[edit services analytics]
user@switch# run show configuration
services {
  analytics {
    export-profiles {
      ep {
        stream-format gpb;
        interface {
          information;
          statistics {
            traffic;
            queue;
          }
        }
      }
    }
  }
}
```

```

    }
    status {
        link;
    }
}
system {
    information;
    status {
        traffic;
        queue;
    }
}
}
}
resource-profiles {
    sys-rp {
        queue-monitoring;
        traffic-monitoring;
        depth-threshold high 99999 low 99;
    }
    if-rp {
        queue-monitoring;
        traffic-monitoring;
        latency-threshold high 2300 low 20;
    }
}
resource {
    system {
        resource-profile sys-rp;
        polling-interval {
            traffic-monitoring 5;
            queue-monitoring 1000;
        }
    }
    interfaces {
        xe-0/0/16 {
            resource-profile if-rp;
        }
        xe-0/0/18 {
            resource-profile if-rp;
        }
        xe-0/0/19 {
            resource-profile if-rp;
        }
    }
}
collector {
    local {
        file an size 10m files 3;
    }
    address 10.94.184.25 {
        port 50013 {
            transport udp {
                export-profile ep;
            }
        }
    }
}

```

```

    }
    address 10.94.198.11 {
        port 50001 {
            transport tcp {
                export-profile ep;
            }
        }
    }
}
}
}
}

```

Verification

Confirm that the configuration is correct and works as expected by performing these tasks:

- [Verifying the Network Analytics Configuration on page 54](#)
- [Verifying the Network Analytics Status on page 54](#)
- [Verifying the Collector Configuration on page 55](#)
- [Verifying Queue Statistics on page 56](#)
- [Verifying Traffic Statistics on page 56](#)

Verifying the Network Analytics Configuration

| | | | | | | |
|----------------|---|--|--|--|--|--|
| Purpose | Verify the configuration for network analytics. | | | | | |
| Action | From operational mode, enter the show analytics configuration command to display the traffic and queue monitoring configuration. <pre> user@host> show analytics configuration Traffic monitoring status is enabled Traffic monitoring polling interval : 5 seconds Queue monitoring status is enabled Queue monitoring polling interval : 1000 milliseconds Queue depth high threshold : 99999 bytes Queue depth low threshold : 99 bytes </pre> | | | | | |
| Meaning | The output displays the traffic and queue monitoring configuration information on the switch. | | | | | |

Verifying the Network Analytics Status

| | |
|----------------|--|
| Purpose | Verify the network analytics operational status of the switch. |
|----------------|--|

Action From operational mode, enter the **show analytics status global** command to display global traffic and queue monitoring status.

```
user@host> show analytics status global
Traffic monitoring status is enabled
Traffic monitoring polling interval : 5 seconds
Queue monitoring status is enabled
Queue monitoring polling interval : 1000 milliseconds
Queue depth high threshold : 99999 bytes
Queue depth low threshold : 99 bytes
```

From operational mode, enter the **show analytics status** command to display both the interface and global queue monitoring status.

```
user@host> show analytics status
Traffic monitoring status is enabled
Traffic monitoring polling interval : 5 seconds
Queue monitoring status is enabled
Queue monitoring polling interval : 1000 milliseconds
Queue depth high threshold : 99999 bytes
Queue depth low threshold : 99 bytes
```

| Interface | Traffic Statistics | Queue Statistics | Queue depth threshold | | Latency threshold | |
|-----------|-----------------------|---------------------|--------------------------|-----|----------------------|-----|
| | | | High | Low | High | Low |
| | | | (bytes) | | (nanoseconds) | |
| xe-0/0/16 | enabled | enabled | n/a | n/a | 2300 | 20 |
| xe-0/0/18 | enabled | enabled | n/a | n/a | 2300 | 20 |
| xe-0/0/19 | enabled | enabled | n/a | n/a | 2300 | 20 |

Meaning The output displays the global and interface status of traffic and queue monitoring on the switch.

Verifying the Collector Configuration

Action Verify the configuration for the collector for streamed data is working.

From operational mode, enter the **show analytics collector** command to display the streaming servers configuration.

```
user@host> show analytics collector
Address      Port  Transport  Stream format  State      Sent
10.94.184.25 50013 udp        gpb            n/a        484
10.94.198.11 50001 tcp        gpb            In progress  0
```

Meaning The output displays the collector configuration.



NOTE: The connection state of a port configured with the **udp** transport protocol is always displayed as **n/a**.

Verifying Queue Statistics

Purpose Verify that queue statistics collection is working.

Action From operational mode, enter the **show analytics queue-statistics** command to display the queue statistics.

```
user@host> show analytics queue-statistics
CLI issued at 2014-03-04 15:37:03.116018
Time                Interface  Queue-depth  Latency
                   (bytes)      (nanoseconds)
00:00:00.412371 ago  xe-0/0/19  1384656      1107724
00:00:01.412395 ago  xe-0/0/19  1375712      1100569
00:00:02.415366 ago  xe-0/0/19  1385280      1108224
00:00:03.417395 ago  xe-0/0/19  1381744      1105395
00:00:04.411392 ago  xe-0/0/19  1368432      1094745
00:00:05.414387 ago  xe-0/0/19  1374880      1099904
00:00:06.414365 ago  xe-0/0/19  1373632      1098905
00:00:07.416386 ago  xe-0/0/19  1370096      1096076
00:00:08.413384 ago  xe-0/0/19  1377168      1101734
00:00:09.415379 ago  xe-0/0/19  1370720      1096576
00:00:10.418374 ago  xe-0/0/19  1381120      1104896
00:00:11.410376 ago  xe-0/0/19  1383408      1106726
00:00:12.412372 ago  xe-0/0/19  1382576      1106060
00:00:13.417371 ago  xe-0/0/19  1387152      1109721
00:00:14.411368 ago  xe-0/0/19  1375296      1100236
---(more)---
```

Meaning The output displays queue-statistics information, with the latest record at the top of the report.

Verifying Traffic Statistics

Purpose Verify that traffic statistics collection is working.

Action From operational mode, enter the **show analytics traffic-statistics** command to display the traffic statistics.

```

user@host> show analytics traffic-statistics
CLI issued at 2014-03-04 15:37:52.047136
Time: 00:00:02.252377 ago, Physical interface: xe-0/0/19
Traffic Statistics:
  Receive          Transmit
Total octets:      15044882432      1502607382656
Total packets:     117538143       11739120146
Unicast packet:    117538143       11739120146
Multicast packets: 0
Broadcast packets: 0
Octets per second: 86488360       8649309384
Packets per second: 84461        8446590
CRC/Align errors:  0
Packets dropped:   0       11760298455
Time: 00:00:02.252377 ago, Physical interface: xe-0/0/18
Traffic Statistics:
  Receive          Transmit
Total octets:      1504619929836    15782818944
Total packets:     11754843131     123303273
Unicast packet:    11754843131     123303273
Multicast packets: 0
Broadcast packets: 0
Octets per second: 8649134008     86487816
Packets per second: 8446458      84461
CRC/Align errors:  5
Packets dropped:   0
Time: 00:00:02.252377 ago, Physical interface: xe-0/0/16
Traffic Statistics:
  Receive          Transmit
Total octets:      1504801437048    757345408
Total packets:     11756261156     5916761
Unicast packet:    11756261156     5916761
Multicast packets: 0
Broadcast packets: 0
Octets per second: 7910619496      0
Packets per second: 7725214      0
CRC/Align errors:  3
Packets dropped:   0

```

Meaning The output displays traffic-statistics information.

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - *analytics*
 - *show analytics status*
 - *show analytics collector*

Configuring Queue Monitoring

Network analytics queue monitoring provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. You can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed.

You enable queue monitoring by first defining a resource profile template, and then applying the profile to the system (for a global configuration) or to individual interfaces.



NOTE: You can configure queue monitoring on physical network interfaces only; logical interfaces and Virtual Chassis physical (VCP) interfaces are not supported.



NOTE: This procedure requires Junos OS Release 13.2X51-D15 or later to be installed on your device.

To configure queue monitoring on a QFX Series standalone switch:

1. Configure the queue monitoring polling interval (in milliseconds) globally (for the system):

[edit]

set services analytics resource system polling-interval queue-monitoring *interval*

2. Configure a resource profile for the system, and enable queue monitoring:

[edit]

set services analytics resource-profiles *profile-name* queue-monitoring

3. Configure high and low values of the depth-threshold (in bytes) for queue monitoring in the system profile:

[edit]

set services analytics resource-profiles *profile-name* depth-threshold high *number* low *number*

For both high and low values, the range is from 1 to 1,250,000,000 bytes, and the default value is 0 bytes.



NOTE: You can configure either the depth-threshold or latency threshold for the system, but not both.

4. Apply the resource profile template to the system for a global configuration:

[edit]

set services analytics resource system resource-profile *profile-name*

5. Configure an interface-specific resource profile and enable queue monitoring for the interface:

```
[edit]
set services analytics resource-profiles profile-name queue-monitoring
```

6. Configure the latency-threshold (high and low values) for queue monitoring in the interface-specific profile:

```
[edit]
set services analytics resource-profiles profile-name latency-threshold high number
low number
```

For both high and low values, the range is from 1 to 100,000,000 nanoseconds, and the default value is 1,000,000 nanoseconds.



NOTE: You can configure either the depth-threshold or latency threshold for interfaces, but not both.

7. Apply the resource profile template for interfaces to one or more interfaces:

```
[edit]
set services analytics resource interfaces interface-name resource-profile profile-name
```



NOTE: If a conflict arises between the system and interface configurations, the interface-specific configuration supersedes the global (system) configuration.

Related Documentation

- [Network Analytics Overview on page 29](#)
- [Example: Configuring Enhanced Network Analytics Features on page 48](#)
- [analytics](#)

Configuring Traffic Monitoring

Network analytics queue monitoring provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. You can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed.

You enable traffic monitoring by first defining a resource profile template, and then applying the profile to the system (for a global configuration) or to individual interfaces.



NOTE: You can configure traffic monitoring on physical network interfaces only; logical interfaces and Virtual Chassis physical (VCP) interfaces are not supported.



NOTE: This procedure requires Junos OS Release 13.2X51-D15 or later to be installed on your device.

To configure traffic monitoring on a QFX Series standalone switch:

1. Configure the traffic monitoring polling interval (in seconds) for the system:

[edit]

set services analytics resource system polling-interval traffic-monitoring *interval*

2. Configure a resource profile for the system, and enable traffic monitoring in the profile:

[edit]

set services analytics resource-profiles *profile-name* traffic-monitoring

3. Apply the resource profile to the system for a global configuration:

[edit]

set services analytics resource system resource-profile *profile-name*

4. Configure a resource profile for interfaces, and enable traffic monitoring in the profile:

[edit]

set services analytics resource-profiles *profile-name* traffic-monitoring



NOTE: If a conflict arises between the system and interface configurations, the interface-specific configuration supersedes the global (system) configuration.

5. Apply the resource profile template to one or more interfaces:

[edit]

set services analytics resource interfaces *interface-name* resource-profile *profile-name*

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - [Example: Configuring Enhanced Network Analytics Features on page 48](#)
 - [analytics](#)

Configuring a Local File for Network Analytics Data

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. Network administrators can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed.

To save the queue and traffic statistics data in a local file, you must configure a filename to store it.



NOTE: This procedure requires Junos OS Release 13.2X51-D15 or later to be installed on your device.

To configure a local file for storing queue and traffic monitoring statistics:

1. Configure a filename:

```
[edit]
set services analytics collector local file filename
```

There is no default filename. If you do not configure a filename, network analytics statistics are not saved locally.

2. Configure the number of files (from 2 to 1000 files):

```
[edit]
set services analytics collector local file filename files number
```

3. Configure the file size (from 10 to 4095 MB) in the format of xm:

```
[edit]
set services analytics collector local file an size size
```

- Related Documentation**
- [Network Analytics Overview on page 29](#)
 - [Example: Configuring Enhanced Network Analytics Features on page 48](#)
 - [analytics](#)

Configuring a Remote Collector for Streaming Analytics Data

The network analytics feature provides visibility into the performance and behavior of the data center infrastructure. This feature collects data from the switch, analyzes the data using sophisticated algorithms, and captures the results in reports. Network administrators can use the reports to help troubleshoot problems, make decisions, and adjust resources as needed.

You can configure an export profile to define the stream format and type of data, and one or more remote servers (collectors) to receive streaming network analytics data.



NOTE: This procedure requires Junos OS Release 13.2X51-D15 or later to be installed on your device.

To configure a collector for receiving streamed analytics data:

1. Create an export profile and specify the stream format:

```
[edit]
set services analytics export-profiles profile-name stream-format format
```

2. Configure the export profile to include interface information:

```
[edit]
set services analytics export-profiles profile-name interface information
```

3. Configure the export profile to include interface queue statistics:

```
[edit]
set services analytics export-profiles profile-name interface statistics queue
```

4. Configure the export profile to include interface traffic statistics:

```
[edit]
set services analytics export-profiles profile-name interface statistics traffic
```

5. Configure the export profile to include interface status link information:

```
[edit]
set services analytics export-profiles profile-name interface status link
```

6. Configure the export profile to include system information:

```
[edit]
set services analytics export-profiles profile-name system information
```

7. Configure the export profile to include system queue status:

```
[edit]
set services analytics export-profiles profile-name system status queue
```

8. Configure the export profile to include system traffic status:

```
[edit]
set services analytics export-profiles profile-name system status traffic
```

9. Configure the transport protocol for the collector addresses and apply the export profile:

[edit]

```
set services analytics collector address ip-address port port transport protocol
export-profile profile-name
set services analytics collector address ip-address port port transport protocol
export-profile profile-name
```



NOTE: If you configure the `tcp` or `udp` option for the JSON, CSV, and TSV formats, you must also set up the TCP or UDP client software on the remote collector to process records that are separated by the newline character (`\n`) on the remote server.

If you configure the `tcp` or `udp` option for the GPB format, you must also set up the TCP or UDP build streaming server using the `analytics.proto` file.

**Related
Documentation**

- [Network Analytics Overview on page 29](#)
- [Example: Configuring Enhanced Network Analytics Features on page 48](#)
- [analytics](#)

CHAPTER 5

Configuring sFlow Technology

- [Understanding How to Use sFlow Technology for Network Monitoring on a Switch on page 65](#)
- [Configuring sFlow Technology on page 70](#)

Understanding How to Use sFlow Technology for Network Monitoring on a Switch

The sFlow technology is a monitoring technology for high-speed switched or routed networks. sFlow monitoring technology randomly samples network packets and sends the samples to a monitoring station called a *collector*. You can configure sFlow technology on a Juniper Networks switch to continuously monitor traffic at wire speed on all interfaces simultaneously.

This topic describes:

- [Sampling Mechanism and Architecture of sFlow Technology on Switches on page 65](#)
- [Adaptive Sampling on page 67](#)
- [sFlow Agent Address Assignment on page 68](#)
- [sFlow Limitations on Switches on page 68](#)

Sampling Mechanism and Architecture of sFlow Technology on Switches

sFlow technology uses the following two sampling mechanisms:

- **Packet-based sampling**—Samples one packet out of a specified number of packets from an interface enabled for sFlow technology. Only the first 128 bytes of each packet are sent to the collector. Data collected include the Ethernet, IP, and TCP headers, along with other application-level headers (if present). Although this type of sampling might not capture infrequent packet flows, the majority of flows are reported over time, allowing the collector to generate a reasonably accurate representation of network activity. To configure packet-based sampling, you must specify a sample rate.
- **Time-based sampling**—Samples interface statistics at a specified interval from an interface enabled for sFlow technology. Statistics such as Ethernet interface errors are captured. To configure time-based sampling, you must specify a polling interval.

The sampling information is used to create a network traffic visibility picture. The Juniper Networks Junos operating system (Junos OS) fully supports the sFlow standard described

in RFC 3176, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks* (see <http://faqs.org/rfcs/rfc3176.html>).



NOTE: sFlow technology on the switches samples only raw packet headers. A raw Ethernet packet is the complete Layer 2 network frame.

An sFlow monitoring system consists of an sFlow agent embedded in the switch and a centralized collector. The sFlow agent's two main activities are random sampling and statistics gathering. It combines interface counters and flow samples and sends them across the network to the sFlow collector as UDP datagrams, directing those datagrams to the IP address and UDP destination port of the collector. Each datagram contains the following information:

- The IP address of the sFlow agent
- The number of samples
- The interface through which the packets entered the agent
- The interface through which the packets exited the agent
- The source and destination interface for the packets
- The source and destination VLAN for the packets

EX Series switches, QFX Series switches, and the QFabric systems adopt the distributed sFlow architecture. The sFlow agent has two separate sampling entities that are associated with each Packet Forwarding Engine in case of switches and nodes in case of a QFabric system. These sampling entities are known as subagents. Each subagent has a unique ID that is used by the collector to identify the data source. A subagent has its own independent state and forwards its own sample messages to the sFlow agent. The sFlow agent is responsible for packaging the samples into datagrams and sending them to the sFlow collector. Because sampling is distributed across subagents, the protocol overhead associated with sFlow technology is significantly reduced at the collector.



NOTE: On the QFabric system, an sFlow collector must be reachable through the data network. Because each Node device has all routes stored in the default routing instance, the collector IP address should be included in the default routing instance to ensure the collector's reachability from the Node device.



NOTE: You cannot configure sFlow monitoring on a link aggregation group (LAG), but you can configure it individually on a LAG member interface.

Infrequent sampling flows might not be reported in the sFlow information, but over time the majority of flows are reported. Based on a configured sampling rate N , 1 out of N packets is captured and sent to the collector. This type of sampling does not provide a

100 percent accurate result in the analysis, but it does provide a result with quantifiable accuracy. A user-configured polling interval defines how often the sFlow data for a specific interface are sent to the collector, but an sFlow agent can also schedule polling.



NOTE: We recommend that you configure the same sample rate for all the ports in a line card. If you configure different sample rates, the lowest value is used for all ports on the line card..



NOTE: If the mastership assignment changes in a Virtual Chassis setup, sFlow technology continues to function.

Adaptive Sampling

To ensure sampling accuracy and efficiency, EX Series switches and QFX Series devices use adaptive sFlow sampling. Adaptive sampling monitors the overall incoming traffic rate on the device and provides feedback to the interfaces to dynamically adapt their sampling rate to traffic conditions. The sFlow agent reads the statistics on the interfaces every few seconds (12 seconds for EX Series switches and 5 seconds for QFX Series devices) and identifies five interfaces with the highest number of samples.

On a Flexible PIC Concentrator (FPC), when the CPU processing limit is reached because of sflow sample processing, a binary backoff algorithm is initiated. This reduces the sampling load, arriving through the top five sample-producing interfaces on that FPC by half. The backoff algorithm achieves this by doubling the sampling rate on these five earmarked interfaces. This process is repeated until the CPU-load due to sflow on the given FPC comes down to an acceptable level.

On a QFabric system, sFlow technology monitors the interfaces on each node device as a group, and implements the binary backoff algorithm based on the traffic on that group of interfaces.



NOTE: On the QFX Series standalone switches, if you configure sFlow technology monitoring on multiple interfaces and with a high sampling rate, we recommend that you specify a collector that is on the data network instead of on the management network. Having a high volume of sFlow technology monitoring traffic on the management network might interfere with other management interface traffic.

Using adaptive sampling prevents overloading of the CPU and keeps the device operating at its optimum level even when there is a change in traffic patterns on the interfaces. The reduced sampling rate is used until the device is rebooted or when a new sampling rate is configured.



NOTE: sFlow technology on EX Series switches does not support graceful restart. When a graceful restart occurs, the adaptive sampling rate is set to the user-configured sampling rate.

sFlow Agent Address Assignment

The sFlow collector uses the sFlow agent's IP address to determine the source of the sFlow data. You can configure the IP address of the sFlow agent to ensure that the agent ID of the sFlow agent remains constant. If you do not specify the IP address to be assigned to the agent, an IP address is automatically assigned to the agent based on the following order of priority of interfaces configured on the device:

| EX Series Devices | QFX Series Devices |
|--|--|
| <ol style="list-style-type: none"> 1. Virtual Management Ethernet (VME) interface 2. Management Ethernet interface | <ol style="list-style-type: none"> 1. Management Ethernet interface me0 IP address 2. Any Layer 3 interface if the me0 IP address is not available |

If a particular interface is not configured, the IP address of the next interface in the priority list is used as the IP address for the agent. Once an IP address is assigned to the agent, the agent ID is not modified until the sFlow service is restarted. At least one interface has to be configured for an IP address to be assigned to the agent. When the agent's IP address is assigned automatically, the IP address is dynamic and changes when the switch reboots.

On the QFabric system, the following default values are used if the optional parameters are not configured:

- Agent ID is the management IP address of the default partition.
- Source IP is the management IP address of the default partition.

In addition, the QFabric system subagent ID (which is included in the sFlow datagrams) is the ID of the node group from which the datagram is sent to the collector.

sFlow data can be used to provide network traffic visibility information. You can explicitly configure the source IP address to be assigned to the sFlow datagrams. If you do not explicitly configure the IP address, the IP address of any of the configured Layer 3 network interfaces is used as the source IP address. If a Layer 3 IP address is not configured, then the agent IP address is used as the source IP address.

sFlow Limitations on Switches

On the QFX Series, limitations of sFlow traffic sampling include the following:

- sFlow sampling on ingress interfaces does not capture CPU-bound traffic.
- sFlow sampling on egress interfaces does not support broadcast and multicast packets.
- Egress samples do not contain modifications made to the packet in the egress pipeline.

- If a packet is discarded because of a firewall filter, the reason code for discarding the packet is not sent to the collector.
- The out-priority field for a VLAN is always set to 0 (zero) on ingress and egress samples.
- On QFX5100 standalone switches and the QFX Series Virtual Chassis (including mixed QFX Series Virtual Chassis), egress firewall filters are not applied to sFlow sampling packets. On these platforms, the software architecture is different from that on other QFX Series devices—sFlow packets are sent by the Routing Engine (not the line card on the host) and do not transit the switch. Egress firewall filters affect data packets that are transiting a switch, but do not affect packets sent by the Routing Engine. As a result, sFlow sampling packets are always sent to the sFlow collector.

EX9200 switches support configuration of only one sampling rate (inclusive of ingress and egress rates) on an FPC. To support compatibility with the sflow configuration of other Juniper Networks products, EX9200 switches still accept multiple rate configuration on different interfaces of the same FPC. However, the switch programs the lowest rate as the sampling rate for all the interfaces of that FPC. The sFlow show command (**show sflow interfaces**) displays the configured rate and the actual (effective) rate. However, different rates on different FPCs is still supported on EX9200 switches.

**Related
Documentation**

- *Example: Monitoring Network Traffic Using sFlow Technology*
- *Example: Configuring sFlow Technology to Monitor Network Traffic on EX Series Switches*
- [Configuring sFlow Technology on page 70](#)
- *Configuring sFlow Technology for Network Monitoring (CLI Procedure)*
- *Monitoring Interface Status and Traffic*

Configuring sFlow Technology

The sFlow technology is a monitoring technology for high-speed switched or routed networks. sFlow monitoring technology collects samples of network packets and sends them in a UDP datagram to a monitoring station called a *collector*. You can configure sFlow technology on a device to monitor traffic continuously at wire speed on all interfaces simultaneously. You must enable sFlow monitoring on each interface individually; you cannot globally enable sFlow monitoring on all interfaces with a single configuration statement. Junos OS fully supports the sFlow technology standard described in RFC 3176, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*.

On the QFabric system, the sFlow monitoring global configuration that is defined on the Director device is distributed to Node groups that have sFlow sampling configured on the interfaces.

To configure sFlow features using the CLI:

1. Configure the IP address and UDP port of at least one collector:

```
[edit protocols sflow]
user@host# set collector ip-address udp-port port-number
```

The default UDP port assigned is 6343.

2. Enable sFlow technology on a specific interface:

```
[edit protocols sflow]
user@host# set interfaces interface-name
```



NOTE: You cannot enable sFlow technology on a Layer 3 VLAN-tagged interface.

You cannot enable sFlow technology on a LAG interface (for example ae0), but you can enable sFlow technology on the member interfaces of the LAG (for example, xe-0/0/1).

3. Specify how often (in seconds) the sFlow agent polls all interfaces at the global level:

```
[edit protocols sflow]
user@host# set polling-interval seconds
```



NOTE: Specify 0 if you do not want to poll the interface.

4. Specify the rate at which packets are sampled at the global level. For example, configuring a *number* of 1000 sets a sample rate of 1 in 1000 packets.

```
[edit protocols sflow]
user@host# set sample-rate number
```

5. (Optional) You can also configure the polling interval and sample rate at the interface level:

```
[edit protocols sflow]
```

```
user@host# set interfaces interface-name polling-interval seconds sample-rate number
```



NOTE: The interface-level configuration overrides the global configuration for the specified interface.

**Related
Documentation**

- *Example: Monitoring Network Traffic Using sFlow Technology*
- *Overview of sFlow Technology*

CHAPTER 6

Configuring SNMP

- [Understanding the Implementation of SNMP on page 74](#)
- [Utility MIB on page 76](#)
- [SNMPv3 Overview on page 77](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78](#)
- [Understanding RMON on page 79](#)
- [RMON MIB Event, Alarm, Log, and History Control Tables on page 81](#)
- [Understanding Health Monitoring on page 83](#)
- [SNMP MIBs Support on page 84](#)
- [SNMP Traps Support on page 100](#)
- [Configuring SNMP on page 113](#)
- [Configuring the SNMP Community String on page 117](#)
- [Configuring SNMP Trap Groups on page 117](#)
- [Adding a Group of Clients to an SNMP Community on page 118](#)
- [Configuring the Interfaces on Which SNMP Requests Can Be Accepted on page 120](#)
- [Configuring MIB Views on page 120](#)
- [Configuring RMON Alarms and Events on page 121](#)
- [Configuring Health Monitoring on page 123](#)
- [Creating SNMPv3 Users on page 124](#)
- [Configuring Access Privileges for a Group on page 125](#)
- [Assigning a Security Name to a Group on page 126](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 127](#)
- [Configuring SNMP Informs on page 128](#)

Understanding the Implementation of SNMP

The QFX Series products support the Simple Network Management Protocol (SNMP) that is implemented in the Junos OS software.



NOTE: By default, SNMP is not enabled on devices running Junos OS. For information on enabling SNMP on a device running Junos OS, see [“Configuring SNMP” on page 113](#).

A typical SNMP implementation includes the following components:

- **Network management system (NMS)**—The NMS is a combination of hardware and software that is used to monitor and administer a network. Software running on the NMS includes the SNMP manager, which collects information about network connectivity, activity, and events by polling the managed devices.
- **Managed device**—A managed device (also called a network element) is any device managed by the NMS. Routers and switches are common examples of managed devices. The SNMP agent is the SNMP process that resides on the managed device and communicates with the NMS.
- **SNMP agent**—The SNMP agent exchanges network management information with SNMP manager software running on an NMS, or host. The agent responds to requests for information and actions from the manager. The agent also controls access to the agent's MIB, the collection of objects that can be viewed or changed by the SNMP manager.

SNMP data is stored in a highly structured, hierarchical format known as a management information base (MIB). The MIB structure is based on a tree structure, which defines a grouping of objects into related sets. Each object in the MIB is associated with an object identifier (OID), which names the object. The “leaf” in the tree structure is the actual managed object instance, which represents a resource, event, or activity that occurs in your network device. The SNMP implementation in Junos OS uses both standard (developed by IETF and documented in RFCs) and Juniper Networks enterprise-specific MIBs.

Communication between the agent and the manager occurs in one of the following forms:

- **Get, GetBulk, and GetNext requests**—The manager requests information from the agent; the agent returns the information in a **Get** response message.
- **Set requests**—The manager changes the value of a MIB object controlled by the agent; the agent indicates status in a **Set** response message.
- **Traps notification**—The agent sends traps to notify the manager of significant events that occur on the network device.

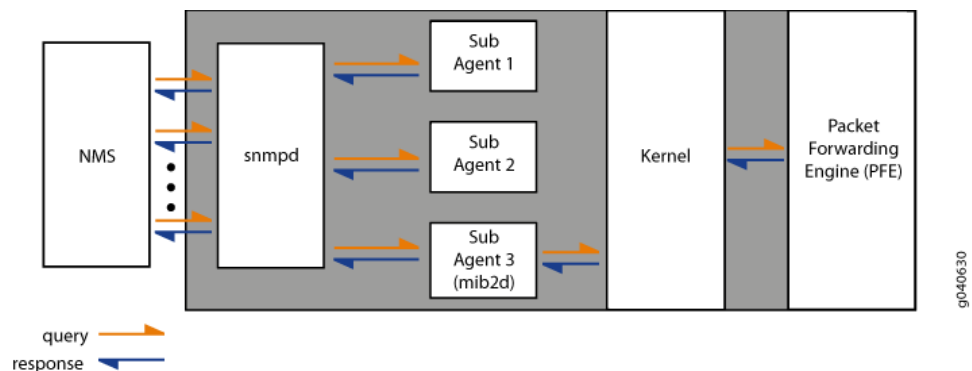
The processes maintaining the SNMP management data include:

- A master SNMP agent (known as SNMP process, or `snmpd`) that resides on the managed device and is managed by the NMS or host.
- Various subagents that reside on different modules of Junos OS, such as the Routing Engine, and are managed by the master SNMP agent.
- Junos OS processes that share data with the subagents when polled for SNMP data (for example, interface-related MIBs).

When an NMS polls the master agent for data, the master agent immediately shares the data with the NMS if the requested data is available from the master agent or one of the subagents. However, if the requested data is not maintained by the master agent or subagents, the subagent polls the Junos OS kernel or the process that maintains that data. The Junos OS kernel may need to get the data from the Packet Forwarding Engine. On receiving the required data, the subagent passes the response back on to the master agent, which in turn passes it on to the NMS.

Figure 6 on page 75 shows the communication flow among the NMS, SNMP master agent (`snmpd`), SNMP subagents, Junos OS kernel, and Packet Forwarding Engine.

Figure 6: SNMP Communication Flow



When a significant event, most often an error or a failure, occurs on a network device, the SNMP agent sends notifications to the SNMP manager. SNMP notifications can be sent as traps (unconfirmed notifications) or inform requests (confirmed notifications).

Junos OS supports trap queuing to ensure that traps are not lost because of temporary unavailability of routes. Two types of queues, destination queues and a throttle queue, are formed to ensure delivery of traps and control the trap traffic. On QFX Series products, the maximum size of trap queues (throttle queue plus destination queue) is 40,960 traps. The maximum size of any one queue is 20,480 traps.

Junos OS forms a destination queue when a trap to a particular destination is returned because the host is not reachable, and it adds the subsequent traps to the same destination to the queue. Junos OS checks for availability of routes every 30 seconds, and sends the traps from the destination queue in a round-robin fashion.

If the trap delivery fails, the trap is added back to the queue, and the delivery attempt counter and the next delivery attempt timer for the queue are reset. Subsequent attempts

occur at progressive intervals of 1 minute, 2 minutes, 4 minutes, and 8 minutes. The maximum delay between the attempts is 8 minutes, and the maximum number of attempts is ten. After ten unsuccessful attempts, the destination queue and all the traps in the queue are deleted.

Junos OS also has a throttle mechanism to control the number of traps (throttle threshold) sent during a particular time period (throttle interval). The throttle mechanism ensures consistency in trap traffic, especially when large numbers of traps are generated because of interface status changes. The throttle interval period begins when the first trap arrives at the throttle. All traps within the trap threshold are processed, and the traps beyond the threshold limit are queued. The default throttle threshold is 500 traps, and the throttle interval default is 5 seconds.



NOTE: You cannot configure trap queueing in Junos OS. You cannot view information about trap queues except for what is provided in the system logs.

Related Documentation

- [Configuring SNMP on page 113](#)
- [SNMP MIBs Support on page 84](#)
- [SNMP Traps Support on page 100](#)

Utility MIB

The Juniper Networks enterprise-specific Utility MIB, whose object ID is `{jnxUtilMibRoot 1}`, defines objects for counters, integers, and strings. The Utility MIB contains one table for each of the following five data types:

- 32-bit counters
- 64-bit counters
- Signed integers
- Unsigned integers
- Octet strings

Each data type has an arbitrary ASCII name, which is defined when the data is populated, and a timestamp that shows the last time when the data instance was modified.

For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos15.1x49/topics/reference/mibs/mib-jnx-util.txt.

For information about the enterprise-specific Utility MIB objects, see the following topics:

- *jnxUtilCounter32Table*
- *jnxUtilCounter64Table*
- *jnxUtilIntegerTable*

- *jnxUtilUintTable*
- *jnxUtilStringTable*

**Related
Documentation**

- *Enterprise-Specific SNMP MIBs Supported by Junos OS*
- *Standard SNMP MIBs Supported by Junos OS*
- *Understanding the Implementation of SNMP on the QFabric System*

SNMPv3 Overview

The QFX3500 switch supports SNMP version 3 (SNMPv3). SNMPv3 enhances the functionality of SNMPv1 and SNMPv2c by supporting user authentication and data encryption. SNMPv3 uses the user-based security model (USM) to provide security for SNMP messages, and the view-based access control model (VACM) for user access control.

SNMPv3 features include:

- With USM, the SNMP messages between the SNMP manager and the agent can have the message source authenticated and the data integrity checked. USM reduces messaging delays and message replays by enforcing timeout limits and by checking for duplicate message request IDs.
- VACM complements USM by providing user access control for SNMP queries to the agent. You define access privileges that you wish to extend to a group of one or more users. Access privileges are determined by the security model parameters (**usm**, **v1**, or **v2**) and security level parameters (**authentication**, **privacy**, or **none**). For each security level, you must associate one MIB view for the group. Associating a MIB view with a group grants the read, write, or notify permission to a set of MIB objects for the group.
- You configure security parameters for each user, including the username, authentication type and authentication password, and privacy type and privacy password. The username given to each user is in a format that is dependent on the security model configured for that user.
- To ensure messaging security, another type of username, called the security name, is included in the messaging data that is sent between the local SNMP server and the destination SNMP server. Each user name is mapped to a security name, but the security name is in a format that is independent of the security model.
- Trap entries in SNMPv3 are created by configuring the notify, notify filter, target address, and target parameters. The **notify** statement specifies the type of notification (trap) and contains a single tag that defines a set of target addresses to receive a trap. The notify filter defines access to a collection of trap object identifiers (OIDs). The target address defines the address of an SNMP management application and other attributes used in sending notifications. Target parameters define the message processing and security parameters used in sending notifications to a particular target.

**Related
Documentation**

- [Assigning a Security Name to a Group on page 126](#)

- [Configuring Access Privileges for a Group on page 125](#)
- [Configuring SNMP Informs on page 128](#)
- [Creating SNMPv3 Users on page 124](#)

Minimum SNMPv3 Configuration on a Device Running Junos OS

To configure the minimum requirements for SNMPv3, include the following statements at the `[edit snmp v3]` and `[edit snmp]` hierarchy levels:



NOTE: You must configure at least one view (notify, read, or write) at the `[edit snmp view-name]` hierarchy level.

```
[edit snmp]
view view-name {
  oid object-identifier (include | exclude);
}
[edit snmp v3]
notify name {
  tag tag-name;
}
notify-filter profile-name {
  oid object-identifier (include | exclude);
}
snmp-community community-index {
  security-name security-name;
}
target-address target-address-name {
  address address;
  target-parameters target-parameters-name;
}
target-parameters target-parameters-name {
  notify-filter profile-name;
  parameters {
    message-processing-model (v1 | v2c | v3);
    security-level (authentication | none | privacy);
    security-model (usm | v1 | v2c);
    security-name security-name;
  }
}
usm {
  local-engine {
    user username {
    }
  }
}
vacm {
  access {
    group group-name {
      (default-context-prefix | context-prefix context-prefix) {
        security-model (any | usm | v1 | v2c) {
          security-level (authentication | none | privacy) {
```

```

        notify-view view-name;
        read-view view-name;
        write-view view-name;
    }
}
}
}
}
security-to-group {
    security-model (usm | v1 | v2c) {
        security-name security-name {
            group group-name;
        }
    }
}
}
}

```

Related Documentation

- [Creating SNMPv3 Users on page 124](#)
- [Configuring MIB Views on page 120](#)
- [Defining Access Privileges for an SNMP Group](#)
- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 127](#)
- [Configuring SNMP Informs on page 128](#)
- [Complete SNMPv3 Configuration Statements](#)
- [Example: SNMPv3 Configuration](#)

Understanding RMON

- [RMON Overview on page 79](#)
- [Alarm Thresholds and Events on page 80](#)

RMON Overview

The Junos OS supports the *Remote Network Monitoring* (RMON) MIB (RFC 2819), which allows a management device to monitor the values of MIB objects, or variables, against configured thresholds. When the value of a variable crosses a threshold, an alarm and its corresponding event are generated. The event can be logged and can generate an SNMP trap.

An operational support system (OSS) or a fault-monitoring system can be used to automatically monitor events that track many different metrics, including performance, availability, faults, and environmental data. For example, an administrator might want to know when the internal temperature of a chassis has risen above a configured threshold, which might indicate that a chassis fan tray is faulty, the chassis air flow is impeded, or the facility cooling system in the vicinity of the chassis is not operating normally.

The RMON MIB also defines tables that store various statistics for Ethernet interfaces, including the **etherStatsTable** and the **etherHistoryTable**. The **etherStatsTable** contains cumulative real-time statistics for Ethernet interfaces, such as the number of unicast,

multicast, and broadcast packets received on an interface. The **etherHistoryTable** maintains a historical sample of statistics for Ethernet interfaces. The control of the **etherHistoryTable**, including the interfaces to track and the sampling interval, is defined by the RMON **historyControlTable**.

To enable RMON alarms, you perform the following steps:

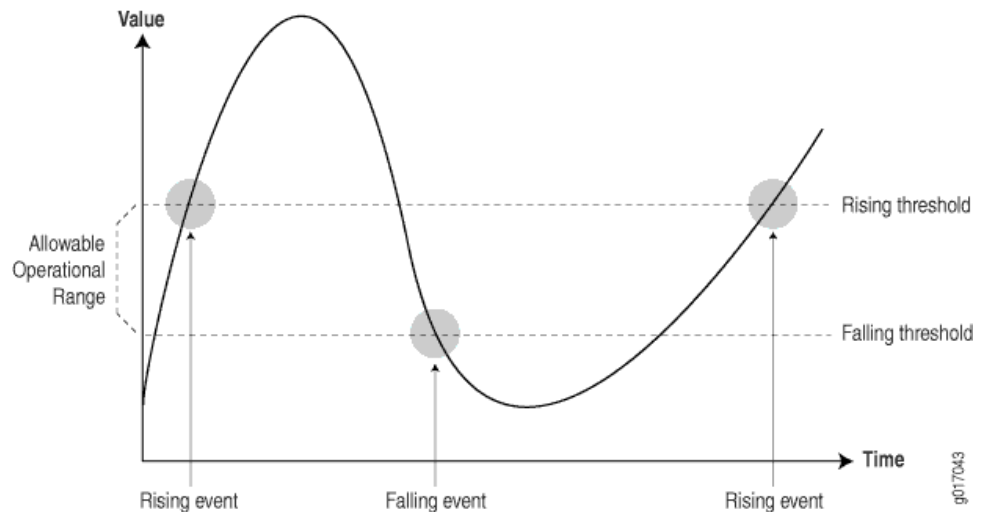
1. Configure SNMP, including trap groups. You configure SNMP at the **[edit snmp]** hierarchy level.
2. Configure rising and falling events in the **eventTable**, including the event types and trap groups. You can also configure events using the CLI at the **[edit snmp rmon event]** hierarchy level.
3. Configure alarms in the **alarmTable**, including the variables to monitor, rising and falling thresholds, the sampling types and intervals, and the corresponding events to generate when alarms occur. You can also configure alarms using the CLI at the **[edit snmp rmon alarm]** hierarchy level.

Extensions to the **alarmTable** are defined in the Juniper Networks enterprise-specific MIB **jnxRmon** (mib-jnx-rmon.txt).

Alarm Thresholds and Events

By setting a rising and a falling threshold for a monitored variable, you can be alerted whenever the value of the variable falls outside the allowable operational range (see [Figure 7 on page 80](#)).

Figure 7: Setting Thresholds



Events are only generated when the alarm threshold is first crossed in any one direction rather than after each sample interval. For example, if a rising threshold alarm, along with its corresponding event, is raised, no more threshold crossing events occur until a corresponding falling alarm occurs. This considerably reduces the quantity of events that are produced by the system, making it easier for operations staff to react when events do occur.

Before you configure remote monitoring, you should identify what variables need to be monitored and their allowable operational range. This requires some period of baselining to determine the allowable operational ranges. An initial baseline period of at least 3 months is not unusual when you first identify the operational ranges and define thresholds, but baseline monitoring should continue over the life span of each monitored variable.

Related Documentation

- [Configuring RMON Alarms and Events on page 121](#)
- [Juniper Networks Enterprise-Specific MIBs](#)
- [RMON MIB Event, Alarm, Log, and History Control Tables on page 81](#)

RMON MIB Event, Alarm, Log, and History Control Tables

The Junos OS supports the *Remote Network Monitoring* (RMON) MIB (RFC 2819), which allows a management device to monitor the values of MIB objects, or variables, against configured thresholds. When the value of a variable crosses a threshold, an alarm and its corresponding event are generated. The event can be logged and can generate an SNMP trap.

[Table 14 on page 81](#) provides each field in the RMON eventTable, the description of the field, and the corresponding Junos OS statement that you can use to configure the field. The Junos OS statements reside at the **[edit snmp rmon]** hierarchy level.

Table 14: RMON Event Table

| Field | Description | Statement [edit snmp rmon] |
|------------------|---|----------------------------|
| eventDescription | Text description of this event. | description |
| eventType | Type of event (for example, log, trap, or log and trap). | type |
| eventCommunity | Trap group to which to send this event, as defined in the Junos OS configuration. (This is not the same as the SNMP community.) | community |
| eventOwner | Entity (for example, manager) that created this event. | — |
| eventStatus | Status of this row (for example, valid, invalid, or createRequest). | — |

[Table 15 on page 81](#) provides each field in the RMON alarmTable, the description of the field, and the corresponding Junos OS statement that you can use to configure the field. The Junos OS statements reside at the **[edit snmp rmon]** hierarchy level.

Table 15: RMON Alarm Table

| Field | Description | Statement [edit snmp rmon] |
|-------------|--|----------------------------|
| alarmStatus | Status of this row (for example, valid, invalid, or createRequest) | — |

Table 15: RMON Alarm Table (*continued*)

| Field | Description | Statement [edit snmp rmon] |
|------------------------|--|----------------------------|
| alarmInterval | Sampling period (in seconds) of the monitored variable | interval |
| alarmVariable | Object identifier (OID) and instance of the variable to be monitored | — |
| alarmValue | Actual value of the sampled variable | — |
| alarmSampleType | Sample type (absolute or delta changes) | sample-type |
| alarmStartupAlarm | Initial alarm (rising, falling, or either) | startup-alarm |
| alarmRisingThreshold | Rising threshold against which to compare the value | rising-threshold |
| alarmFallingThreshold | Falling threshold against which to compare the value | falling-threshold |
| alarmRisingEventIndex | Index (row) of the rising event in the event table | rising-event-index |
| alarmFallingEventIndex | Index (row) of the falling event in the event table | falling-event-index |

Table 16 on page 82 provides each field in the jnxRmon jnxRmonAlarmTable, which is an extension to the RMON alarmTable. You can troubleshoot the RMON agent, rmopd, that runs on a switch by inspecting the contents of the jnxRmonAlarmTable object.

Table 16: jnxRmon Alarm Table

| Field | Description |
|---------------------------|--|
| jnxRmonAlarmGetFailCnt | Number of times the internal Get request for the variable failed |
| jnxRmonAlarmGetFailTime | Value of the sysUpTime object when the last failure occurred |
| jnxRmonAlarmGetFailReason | Reason why the Get request failed |
| jnxRmonAlarmGetOkTime | Value of the sysUpTime object when the variable moved out of failure state |
| jnxRmonAlarmState | Status of this alarm entry |

Table 17 on page 83 provides each field in the RMON historyControlTable, the description of the field, and the corresponding Junos OS statement that you can use to configure the field. The Junos OS statements reside at the **[edit snmp rmon history]** hierarchy level. The historyControlTable controls the RMON etherHistoryTable.

Table 17: RMON History Control Table

| Field | Description | Statement [edit snmp rmon history] |
|--------------------------------|---|------------------------------------|
| historyControlDataSource | Identifies the source of the data for which historical data was collected. | interface |
| historyControlBucketsRequested | Requested number of discrete time intervals over which data is to be saved. | bucket-size |
| historyControlBucketsGranted | Number of discrete sampling intervals over which data is to be saved. | — |
| historyControlInterval | Interval, in seconds, over which the data is sampled for each bucket. | interval |
| historyControlOwner | Entity that configured this entry. | owner |
| historyControlStatus | Status of this entry. | — |

- Related Documentation**
- [Configuring RMON Alarms and Events on page 121](#)
 - [Juniper Networks Enterprise-Specific MIBs](#)
 - [Understanding RMON on page 79](#)

Understanding Health Monitoring

Health monitoring is an SNMP feature that extends the RMON alarm infrastructure to provide monitoring for a predefined set of objects (such as file system usage, CPU usage, and memory usage), and for Junos OS processes.

You enable the health monitor feature using the **health-monitor** statement at the **[edit snmp]** hierarchy level. You can also configure health monitor parameters such as a falling threshold, rising threshold, and interval. If the value of a monitored object exceeds the rising or falling threshold, an alarm is triggered and an event may be logged.

The falling threshold is the lower threshold for the monitored object instance. The rising threshold is the upper threshold for the monitored object instance. Each threshold is expressed as a percentage of the maximum possible value. The interval represents the period of time, in seconds, over which the object instance is sampled and compared with the rising and falling thresholds.

Events are only generated when a threshold is first crossed in any one direction, rather than after each sample interval. For example, if a rising threshold alarm, along with its corresponding event, is raised, no more threshold crossing events occur until a corresponding falling alarm occurs.

System log entries for health monitor events have a corresponding HEALTHMONITOR tag and not a generic SNMPD_RMON_EVENTLOG tag. However, the health monitor sends generic RMON risingThreshold and fallingThreshold traps. You can use the **show snmp**

health-monitor operational command to view information about health monitor alarms and logs.

When you configure the health monitor, monitoring information for certain object instances is available, as shown in [Table 18 on page 84](#).

Table 18: Monitored Object Instances

| Object | Description |
|---------------------------|---|
| jnxHrStoragePercentUsed.1 | Monitors the /dev/ad0s1a : file system on the switch. This is the root file system mounted on / . |
| jnxHrStoragePercentUsed.2 | Monitors the /dev/ad0s1e : file system on the switch. This is the configuration file system mounted on /config . |
| jnxOperatingCPU (RE0) | Monitors CPU usage by the Routing Engine (RE0). |
| jnxOperatingBuffer (RE0) | Monitors the amount of memory available on the Routing Engine (RE0). |
| sysApplElmtRunCPU | Monitors the CPU usage for each Junos OS process (also called daemon). Multiple instances of the same process are monitored and indexed separately. |
| sysApplElmtRunMemory | Monitors the memory usage for each Junos OS process. Multiple instances of the same process are monitored and indexed separately. |

- Related Documentation**
- [Configuring Health Monitoring on page 123](#)
 - *falling-threshold (Health Monitor)*
 - *interval (Health Monitor)*
 - *rising-threshold (Health Monitor)*
 - *show snmp health-monitor*

SNMP MIBs Support

The QFX Series standalone switches, QFX Series Virtual Chassis, and QFabric systems support standard MIBs and Juniper Networks enterprise-specific MIBs.

For more information, see:

- [MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis on page 84](#)
- [MIBs Supported on QFabric Systems on page 94](#)

MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

The QFX Series standalone switches and QFX Series Virtual Chassis support both standard MIBs and Juniper Networks enterprise-specific MIBs. For more information, see:

- [Table 19 on page 85](#) for standard MIBs.

- [Table 20 on page 90](#) for Juniper Networks enterprise-specific MIBs.

Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| RFC | Additional Information |
|---|---|
| IEEE 802.1ab section 12.1, <i>Link Layer Discovery Protocol (LLDP) MIB</i> | Supported tables and objects: <ul style="list-style-type: none"> • lldpRemManAddrOID • lldpLocManAddrOID • lldpReinitDelay • lldpNotificationInterval • lldpStatsRxPortFramesDiscardedTotal • lldpStatsRxPortFramesError • lldpStatsRxPortTLVsDiscardedTotal • lldpStatsRxPortTLVsUnrecognizedTotal • lldpStatsRxPortAgeoutsTotal |
| IEEE 802.3ad, <i>Aggregation of Multiple Link Segments</i> | The following tables and objects are supported: <ul style="list-style-type: none"> • dot3adAggPortTable, dot3adAggPortListTable, dot3adAggTable, and dot3adAggPortStatsTable • dot3adAggPortDebugTable (only dot3adAggPortDebugRxState, dot3adAggPortDebugMuxState, dot3adAggPortDebugActorSyncTransitionCount, dot3adAggPortDebugPartnerSyncTransitionCount, dot3adAggPortDebugActorChangeCount, and dot3adAggPortDebugPartnerChangeCount) • dot3adTablesLastChanged |
| RFC 1155, <i>Structure and Identification of Management Information for TCP/IP-based Internets</i> | — |
| RFC 1157, <i>A Simple Network Management Protocol (SNMP)</i> | — |
| RFC 1212, <i>Concise MIB Definitions</i> | — |
| RFC 1213, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> | The following areas are supported: <ul style="list-style-type: none"> • MIB II and its SNMP version 2 derivatives, including: <ul style="list-style-type: none"> • Statistics counters • IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, <i>IP Forwarding Table MIB</i>) • ipAddrTable • SNMP management • Interface management • SNMPv1 Get, GetNext requests, and SNMPv2 GetBulk request • Junos OS-specific secured access list • Master configuration keywords • Reconfigurations upon SIGHUP |

Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (continued)

| RFC | Additional Information |
|---|--|
| RFC 1215, <i>A Convention for Defining Traps for use with the SNMP</i> | Support is limited to MIB II SNMP version 1 traps and version 2 notifications. |
| RFC 1286, <i>Definitions of Managed Objects for Bridges</i> | — |
| RFC 1657, <i>Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2</i> | — |
| RFC 1850, <i>OSPF Version 2 Management Information Base</i> | The following table, objects, and traps are not supported: <ul style="list-style-type: none"> • Host Table • ospfOriginateNewLsas and ospfRxNewLsas objects • ospfOriginateLSA, ospfLsdbOverflow, and ospfLsdbApproachingOverflow traps |
| RFC 1901, <i>Introduction to Community-based SNMPv2</i> | — |
| RFC 1905, <i>Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | — |
| RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | — |
| RFC 2011, <i>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</i> | — |
| RFC 2012, <i>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</i> | — |
| RFC 2013, <i>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</i> | — |
| RFC 2233, <i>The Interfaces Group MIB Using SMIv2</i> | NOTE: RFC 2233 has been replaced by RFC 2863. However, Junos OS supports both RFC 2233 and RFC 2863. |
| RFC 2287, <i>Definitions of System-Level Managed Objects for Applications</i> | The following objects are supported: <ul style="list-style-type: none"> • sysApplInstallPkgTable • sysApplInstallElmtTable • sysApplElmtRunTable • sysApplMapTable |

Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| RFC | Additional Information |
|--|---|
| RFC 2570, <i>Introduction to Version 3 of the Internet-standard Network Management Framework</i> | — |
| RFC 2571, <i>An Architecture for Describing SNMP Management Frameworks</i> (read-only access) | NOTE: RFC 2571 has been replaced by RFC 3411. However, Junos OS supports both RFC 2571 and RFC 3411. |
| RFC 2572, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> (read-only access) | NOTE: RFC 2572 has been replaced by RFC 3412. However, Junos OS supports both RFC 2572 and RFC 3412. |
| RFC 2576, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i> | NOTE: RFC 2576 has been replaced by RFC 3584. However, Junos OS supports both RFC 2576 and RFC 3584. |
| RFC 2578, <i>Structure of Management Information Version 2 (SMIv2)</i> | — |
| RFC 2579, <i>Textual Conventions for SMIv2</i> | — |
| RFC 2580, <i>Conformance Statements for SMIv2</i> | — |
| RFC 2665, <i>Definitions of Managed Objects for the Ethernet-like Interface Types</i> | — |
| RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> | Support does not include row creation, the Set operation, and the vrrpStatsPacketLengthErrors object. |
| RFC 2790, <i>Host Resources MIB</i> | Support is limited to the following objects: <ul style="list-style-type: none"> Only hrStorageTable. The file systems <code>/</code>, <code>/config</code>, <code>/var</code>, and <code>/tmp</code> always return the same index number. When SNMP restarts, the index numbers for the remaining file systems might change. Only the objects of the hrSystem and hrSWInstalled groups. |
| RFC 2819, <i>Remote Network Monitoring Management Information Base</i> | The following objects are supported: <ul style="list-style-type: none"> etherStatsTable (for Ethernet interfaces only), alarmTable, eventTable, and logTable. historyControlTable and etherHistoryTable (except the etherHistoryUtilization object). |
| RFC 2863, <i>The Interfaces Group MIB</i> | NOTE: RFC 2233 has been replaced by RFC 2863. However, Junos OS supports both RFC 2233 and RFC 2863. |
| RFC 2932, <i>IPv4 Multicast Routing MIB</i> | — |

Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (continued)

| RFC | Additional Information |
|--|---|
| RFC 2933, <i>Internet Group Management Protocol (IGMP) MIB</i> | — |
| RFC 2934, <i>Protocol Independent Multicast MIB for IPv4</i> | In Junos OS, RFC 2934 is implemented based on a draft version, <i>pimmib.mib</i> , of the now standard RFC. |
| RFC 3410, <i>Introduction and Applicability Statements for Internet Standard Management Framework</i> | — |
| RFC 3411, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i> | NOTE: RFC 3411 replaces RFC 2571. However, Junos OS supports both RFC 3411 and RFC 2571. |
| RFC 3412, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3412 replaces RFC 2572. However, Junos OS supports both RFC 3412 and RFC 2572. |
| RFC 3413, <i>Simple Network Management Protocol (SNMP) Applications</i> | All MIBs are supported except for the Proxy MIB. |
| RFC 3414, <i>User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)</i> | — |
| RFC 3415, <i>View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i> | — |
| RFC 3416, <i>Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3416 replaces RFC 1905, which was supported in earlier versions of Junos OS. |
| RFC 3417, <i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i> | — |
| RFC 3418, <i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3418 replaces RFC 1907, which was supported in earlier versions of Junos OS. |
| RFC 3584, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i> | — |
| RFC 3826, <i>The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model</i> | — |

Table 19: Standard MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| RFC | Additional Information |
|--|---|
| RFC 4188, <i>Definitions of Managed Objects for Bridges</i> | <p>The QFX3500 and QFX3600 switches support 802.1D STP (1998) and the following subtrees and objects only:</p> <ul style="list-style-type: none"> dot1dTp subtree—dot1dTpFdbAddress, dot1dTpFdbPort, and dot1dTpFdbStatus objects from the dot1dTpFdbTable table. dot1dBase subtree—dot1dBasePort and dot1dBasePortIfIndex objects from the dot1dBasePortTable table. <p>NOTE: On QFX3500 and QFX3600 switches, the dot1dTpFdbTable table is populated only with MAC addresses learned on the default VLAN. To see the MAC addresses of all VLANs, specify the dot1qTpFdbTable table (RFC 4363b, <i>Q-Bridge VLAN MIB</i>) when you issue the show snmp mib walk command.</p> <p>Not supported on OCX Series devices.</p> |
| RFC 4293, <i>Management Information Base for the Internet Protocol (IP)</i> | Supports the ipAddrTable table only. |
| RFC 4318, <i>Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol</i> | <p>Supports 802.1w and 802.1t extensions for RSTP.</p> <p>Not supported on OCX Series devices.</p> |
| RFC 4363b, <i>Q-Bridge VLAN MIB</i> | <p>NOTE: On QFX3500 and QFX3600 switches, the dot1dTpFdbTable table (RFC 4188, <i>Definitions of Managed Objects for Bridges</i>) is populated only with MAC addresses learned on the default VLAN. To see the MAC addresses of all VLANs, specify the dot1qTpFdbTable table (in this MIB) when you issue the show snmp mib walk command.</p> <p>Not supported on OCX Series devices.</p> |
| RFC 4444, <i>IS-IS MIB</i> | — |
| Internet Assigned Numbers Authority, <i>IANAiftype Textual Convention MIB</i> (referenced by RFC 2233) | See http://www.iana.org/assignments/ianaiftype-mib . |
| Internet draft draft-reeder-snmppv3-usm-3desede-00.txt, <i>Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in 'Outside' CBC Mode</i> | — |
| Internet draft draft-ietf-idmr-igmp-mib-13.txt, <i>Internet Group Management Protocol (IGMP) MIB</i> | — |
| ESO Consortium MIB | <p>NOTE: The ESO Consortium MIB has been replaced by RFC 3826. See http://www.snmp.com/eso/.</p> |

Table 20: Juniper Networks Enterprise-Specific MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| MIB | Description |
|---|--|
| Alarm MIB (mib-jnx-chassis-alarm) | <p>Provides support for alarms from the switch.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis-alarm.txt.</p> <p>For more information, see <i>Alarm MIB</i>.</p> |
| Analyzer MIB (mib-jnx-analyzer) | <p>Contains analyzer and remote analyzer data related to port mirroring.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-analyzer.txt.</p> <p>For more information, see <i>Analyzer MIB</i>.</p> <p>Not supported on OCX Series devices.</p> |
| Chassis MIB (mib-jnx-chassis) | <p>Provides support for environmental monitoring (power supply state, board voltages, fans, temperatures, and airflow) and inventory support for the chassis, Flexible PIC Concentrators (FPCs), and PICs.</p> <p>NOTE: The jnxLEDTable table has been deprecated.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chassis.txt.</p> <p>For more information, see <i>Chassis MIBs</i>.</p> |
| Chassis Definitions for Router Model MIB (mib-jnx-chas-defines) | <p>Contains the object identifiers (OIDs) that are used by the Chassis MIB to identify routing and switching platforms and chassis components. The Chassis MIB provides information that changes often, whereas the Chassis Definitions for Router Model MIB provides information that changes less often.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-chas-defines.txt.</p> <p>For more information, see <i>Chassis MIBs</i>.</p> |
| Class-of-Service MIB (mib-jnx-cos) | <p>Provides support for monitoring interface output queue statistics per interface and per forwarding class.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cos.txt.</p> <p>For more information, see <i>Class-of-Service MIB</i>.</p> |

Table 20: Juniper Networks Enterprise-Specific MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| MIB | Description |
|---|---|
| Configuration Management MIB (mib-jnx-cfgmgt) | <p>Provides notification for configuration changes and rescue configuration changes in the form of SNMP traps. Each trap contains the time at which the configuration change was committed, the name of the user who made the change, and the method by which the change was made.</p> <p>A history of the last 32 configuration changes is kept in jnxCmChgEventTable.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cfgmgt.txt.</p> <p>For more information, see <i>Configuration Management MIB</i>.</p> |
| Ethernet MAC MIB (mib-jnx-mac) | <p>Monitors media access control (MAC) statistics on Gigabit Ethernet intelligent queuing (IQ) interfaces. It collects MAC statistics; for example, inoctets, inframes, outoctets, and outframes on each source MAC address and virtual LAN (VLAN) ID for each Ethernet port.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mac.txt.</p> <p>For more information, see <i>Ethernet MAC MIB</i>.</p> <p>Not supported on OCX Series devices.</p> |
| Event MIB (mib-jnx-event) | <p>Defines a generic trap that can be generated using an operations script or event policy. This MIB provides the ability to specify a system log string and raise a trap if that system log string is found.</p> <p>In Junos OS release 13.2X51-D10 or later, if you configured an event policy to raise a trap when a new SNMP trap target is added, the SNMPD_TRAP_TARGET_ADD_NOTICE trap is generated with information about the new target.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-event.txt.</p> <p>For more information, see <i>Event MIB</i>.</p> |
| Firewall MIB (mib-jnx-firewall) | <p>Provides support for monitoring firewall filter counters.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-firewall.txt.</p> <p>For more information, see <i>Firewall MIB</i>.</p> |

Table 20: Juniper Networks Enterprise-Specific MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| MIB | Description |
|---|--|
| Host Resources MIB (mib-jnx-hostresources) | <p>Extends the hrStorageTable object, providing a measure of the usage of each file system on the switch as a percentage. Previously, the objects in the hrStorageTable measured the usage in allocation units—hrStorageUsed and hrStorageAllocationUnits—only. Using the percentage measurement, you can more easily monitor and apply thresholds on usage.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-hostresources.txt.</p> <p>For more information, see <i>Host Resources MIB</i>.</p> |
| Interface MIB (Extensions) (mib-jnx-if-extensions) | <p>Extends the standard ifTable (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information in the ifJnxTable and ifChassisTable tables.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-if-extensions.txt.</p> <p>For more information, see <i>Interface MIB</i>.</p> |
| MPLS MIB (mib-jnx-mpls) | <p>Provides MPLS information and defines MPLS notifications.</p> <p>NOTE: This MIB is not supported on the QFX5100 switch.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls.txt.</p> <p>For more information, see <i>MPLS MIB</i>.</p> |
| MPLS LDP MIB (mib-jnx-mpls-ldp) | <p>Contains object definitions as described in RFC 3815, <i>Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)</i>.</p> <p>NOTE: This MIB is not supported on the QFX5100 switch.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-mpls-ldp.txt.</p> <p>For more information, see <i>MPLS LDP MIB</i>.</p> |
| Ping MIB (mib-jnx-ping) | <p>Extends the standard Ping MIB control table (RFC 2925). Items in this MIB are created when entries are created in pingCtlTable of the Ping MIB. Each item is indexed exactly as it is in the Ping MIB.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-ping.txt.</p> <p>For more information, see <i>PING MIB</i>.</p> |

Table 20: Juniper Networks Enterprise-Specific MIBs Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| MIB | Description |
|---|---|
| RMON Events and Alarms MIB (mib-jnx-rmon) | <p>Supports Junos OS extensions to the standard Remote Monitoring (RMON) Events and Alarms MIB (RFC 2819). The extension augments the alarmTable object with additional information about each alarm. Two additional traps are also defined to indicate when problems are encountered with an alarm.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-rmon.txt.</p> <p>For more information, see <i>RMON Events and Alarms MIB</i>.</p> |
| Structure of Management Information MIB (mib-jnx-smi) | <p>Explains how the Juniper Networks enterprise-specific MIBs are structured.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-smi.txt.</p> <p>For more information, see <i>Structure of Management Information MIB</i>.</p> |
| System Log MIB (mib-jnx-syslog) | <p>Enables notification of an SNMP trap-based application when an important system log message occurs.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-syslog.txt.</p> <p>For more information, see <i>System Log MIB</i>.</p> |
| Utility MIB (mib-jnx-util) | <p>Provides you with SNMP MIB container objects of the following types: 32-bit counters, 64-bit counters, signed integers, unsigned integers, and octet strings. You can use these objects to store data that can be retrieved using other SNMP operations.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-util.txt.</p> <p>For more information, see “Utility MIB” on page 76 and “Using the Enterprise-Specific Utility MIB to Enhance SNMP Coverage” on page 176.</p> |
| VLAN MIB (mib-jnx-vlan) | <p>Contains information about prestandard IEEE 802.10 VLANs and their association with LAN emulation clients.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-vlan.txt.</p> <p>For more information, see <i>VLAN MIB</i>.</p> <p>Not supported on OCX Series devices.</p> |

MIBs Supported on QFabric Systems

The QFabric systems support both standard MIBs and Juniper Networks enterprise-specific MIBs. For more information, see:

- [Table 21 on page 94](#) for standard MIBs.
- [Table 22 on page 98](#) for Juniper Networks enterprise-specific MIBs.

Table 21: Standard MIBs Supported on QFabric Systems

| RFC | Additional Information |
|---|---|
| RFC 1155, <i>Structure and Identification of Management Information for TCP/IP-based Internets</i> | — |
| RFC 1157, <i>A Simple Network Management Protocol (SNMP)</i> | — |
| RFC 1212, <i>Concise MIB Definitions</i> | — |
| RFC 1213, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> | <p>The following areas are supported:</p> <ul style="list-style-type: none"> • MIB II and its SNMP version 2 derivatives, including: <ul style="list-style-type: none"> • Statistics counters • IP, except for ipRouteTable, which has been replaced by ipCidrRouteTable (RFC 2096, <i>IP Forwarding Table MIB</i>) • ipAddrTable • SNMP management • Interface management • SNMPv1 Get, GetNext requests, and version 2 GetBulk request • Junos OS-specific secured access list • Master configuration keywords • Reconfigurations upon SIGHUP |
| RFC 1215, <i>A Convention for Defining Traps for use with the SNMP</i> | Support is limited to MIB II SNMP version 1 traps and version 2 notifications. |
| RFC 1286, <i>Definitions of Managed Objects for Bridges</i> | — |
| RFC 1901, <i>Introduction to Community-based SNMPv2</i> | — |
| RFC 1905, <i>Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | — |
| RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | — |

Table 21: Standard MIBs Supported on QFabric Systems (*continued*)

| RFC | Additional Information |
|--|---|
| RFC 2011, <i>SNMPv2 Management Information Base for the Internet Protocol Using SMIv2</i> | NOTE: On the QFabric system, for the SNMP mibwalk request to work, you must configure the IP address of at least one interface besides the management Ethernet interfaces (me0 and me1) in the Director group. |
| RFC 2012, <i>SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2</i> | — |
| RFC 2013, <i>SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2</i> | — |
| RFC 2233, <i>The Interfaces Group MIB Using SMIv2</i> | <p>NOTE: RFC 2233 has been replaced by RFC 2863. However, Junos OS supports both RFC 2233 and RFC 2863.</p> <p>NOTE: The QFabric system supports the following objects only: ifNumber, ifTable, and ifxTable.</p> |
| RFC 2571, <i>An Architecture for Describing SNMP Management Frameworks</i> (read-only access) | NOTE: RFC 2571 has been replaced by RFC 3411. However, Junos OS supports both RFC 2571 and RFC 3411. |
| RFC 2572, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> (read-only access) | NOTE: RFC 2572 has been replaced by RFC 3412. However, Junos OS supports both RFC 2572 and RFC 3412. |
| RFC 2576, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i> | NOTE: RFC 2576 has been replaced by RFC 3584. However, Junos OS supports both RFC 2576 and RFC 3584. |
| RFC 2578, <i>Structure of Management Information Version 2 (SMIv2)</i> | — |
| RFC 2579, <i>Textual Conventions for SMIv2</i> | — |
| RFC 2580, <i>Conformance Statements for SMIv2</i> | — |

Table 21: Standard MIBs Supported on QFabric Systems (*continued*)

| RFC | Additional Information |
|--|--|
| RFC 2665, <i>Definitions of Managed Objects for the Ethernet-like Interface Types</i> | <p>The QFabric system supports the following tables only:</p> <ul style="list-style-type: none"> dot3StatsTable—There is one row with statistics for each Ethernet-like interface in the QFabric system. The dot3StatsIndex is an interface index that is unique across the system. dot3ControlTable—There is one row in this table for each Ethernet-like interface in the QFabric system that implements the MAC control sublayer. OIDs supported are dot3ControlFunctionsSupported and dot3ControlInUnknownOpcode. dot3PauseTable—There is one row in this table for each Ethernet-like interface in the QFabric system that supports the MAC control PAUSE function. OIDs supported are dot3PauseAdminMode, dot3PauseOperMode, dot3InPauseFrames, and dot3OutPauseFrames. <p>NOTE: Scalar variables are not supported on the QFabric system.</p> |
| RFC 2863, <i>The Interfaces Group MIB</i> | <p>NOTE: RFC 2233 has been replaced by RFC 2863. However, Junos OS supports both RFC 2233 and RFC 2863.</p> <p>NOTE: The QFabric system supports the following objects only: ifNumber, ifTable, and ifxTable.</p> |
| RFC 2933, <i>Internet Group Management Protocol (IGMP) MIB</i> | — |
| RFC 3410, <i>Introduction and Applicability Statements for Internet Standard Management Framework</i> | — |
| RFC 3411, <i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i> | NOTE: RFC 3411 replaces RFC 2571. However, Junos OS supports both RFC 3411 and RFC 2571. |
| RFC 3412, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3412 replaces RFC 2572. However, Junos OS supports both RFC 3412 and RFC 2572. |
| RFC 3416, <i>Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3416 replaces RFC 1905, which was supported in earlier versions of Junos OS. |
| RFC 3417, <i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i> | — |
| RFC 3418, <i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i> | NOTE: RFC 3418 replaces RFC 1907, which was supported in earlier versions of Junos OS. |
| RFC 3584, <i>Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework</i> | — |

Table 21: Standard MIBs Supported on QFabric Systems (*continued*)

| RFC | Additional Information |
|---|---|
| RFC 4188, <i>Definitions of Managed Objects for Bridges</i> | <p>The QFabric system support is limited to the following objects:</p> <ul style="list-style-type: none"> Under the dot1dBase OID, the dot1dBasePortTable table supports only the first two columns in the table: dot1dBasePort and dot1dBasePortIfIndex. The system does not implement the optional traps supporting dot1dNotifications (dot1dBridge 0). Under the dot1dStp OID, supports only the dot1dStpPortTable table. Does not support the scalar variables under dot1dStp. The system does not support scalar variables under dot1dTp, but under that, the dot1dTpFdbTable table is supported (dot1dBridge 4). For OIDs with tables support only, scalar values that are returned by the SNMP agent may not be meaningful and are therefore not recommended for use. <p>Not supported on OCX Series devices.</p> |
| RFC 4293, <i>Management Information Base for the Internet Protocol (IP)</i> | <p>Supports the ipAddrTable table only.</p> <p>On the QFabric system, supported objects in the ipAddrTable table include: ipAdEntAddr, ipAdEntIfIndex, ipAdEntNetMask, ipAdEntBcastAddr, and ipAdEntReasmMaxSize.</p> <p>NOTE: On the QFabric system, for the SNMP mibwalk request to work, you must configure the IP address of at least one interface besides the management Ethernet interfaces (me0 and me1) in the Director group.</p> |
| RFC 4363b, <i>Q-Bridge VLAN MIB</i> | <p>The QFabric system supports the following tables only:</p> <ul style="list-style-type: none"> dot1qTpFdbTable dot1qVlanStaticTable dot1qPortVlanTable dot1qFdbTable <p>Not supported on OCX Series devices.</p> |



NOTE: QFabric-specific MIBs are not supported on OCX Series devices.

Table 22: Juniper Networks Enterprise-Specific MIBs Supported on QFabric Systems

| MIB | Description |
|------------------------------------|--|
| Analyzer MIB (mib-jnx-analyzer) | <p>Contains analyzer and remote analyzer data related to port mirroring.</p> <p>The QFabric system supports:</p> <ul style="list-style-type: none"> Analyzer table—jnxAnalyzerName, jnxMirroringRatio, jnxLossPriority. Analyzer input table—jnxAnalyzerInputValue, jnxAnalyzerInputOption, jnxAnalyzerInputType. Analyzer output table—jnxAnalyzerOutputValue, jnxAnalyzerOutputType. <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-analyzer.txt.</p> <p>For more information, see <i>Analyzer MIB</i>.</p> |
| Chassis MIB (mib-jnx-chassis) | <p>NOTE: The Chassis MIB has been deprecated for the QFabric system. We recommend that you use the Fabric Chassis MIB (mib-jnx-fabric-chassis) for information about the QFabric system.</p> |
| Class-of-Service MIB (mib-jnx-cos) | <p>Provides support for monitoring interface output queue statistics per interface and per forwarding class.</p> <p>The QFabric system supports the following tables and objects:</p> <ul style="list-style-type: none"> Jnxcosifstatflagtable—jnxCosIfstatFlags and jnxCosIfIndex. Jnxcosqstattable—jnxCosQstatTxedPkts, jnxCosQstatTxedPktRate, jnxCosQstatTxedBytes, and jnxCosQstatTxedByteRate. Jnxcosfcidtable—jnxCosFcIdToFcName. Jnxcosfctable—jnxCosFcQueueNr. <p>The QFabric system does not support any traps for this MIB.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cos.txt.</p> <p>For more information, see <i>Class-of-Service MIB</i>.</p> |

Table 22: Juniper Networks Enterprise-Specific MIBs Supported on QFabric Systems (*continued*)

| MIB | Description |
|---|---|
| Configuration Management MIB (mib-jnx-cfgmgmt) | <p>Provides notification for configuration changes and rescue configuration changes in the form of SNMP traps. Each trap contains the time at which the configuration change was committed, the name of the user who made the change, and the method by which the change was made.</p> <p>A history of the last 32 configuration changes is kept in jnxCmChgEventTable.</p> <p>NOTE: On the QFabric system, these conditions apply:</p> <ul style="list-style-type: none"> • All scalar variables under the jnxCmCfgChg table are supported. • Supported scalar OIDs are jnxCmCfgChgLatestIndex, jnxCmCfgChgLatestTime, jnxCmCfgChgLatestDate, jnxCmCfgChgLatestSource, jnxCmCfgChgLatestUser, and jnxCmCfgChgMaxEventEntries. • Scalar variables under the jnxCmRescueChg table are not supported. <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-cfgmgmt.txt.</p> <p>For more information, see <i>Configuration Management MIB</i>.</p> |
| Fabric Chassis MIB (mib-jnx-fabric-chassis) | <p>Provides hardware information about the QFabric system and its component devices. This MIB is based on the Juniper Networks enterprise-specific Chassis MIB but adds another level of indexing that provides information for QFabric system component devices.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-fabric-chassis.txt.</p> <p>For more information, see <i>Fabric Chassis MIB</i>.</p> |
| Interface MIB (Extensions) (mib-jnx-if-extensions) | <p>Extends the standard ifTable (RFC 2863) with additional statistics and Juniper Networks enterprise-specific chassis information in the ifJnxTable and ifChassisTable tables.</p> <p>NOTE: On the QFabric system, scalar variables are not supported.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-if-extensions.txt.</p> <p>For more information, see <i>Interface MIB</i>.</p> |
| Power Supply Unit MIB (mib-jnx-power-supply-unit) | <p>Provides support for environmental monitoring of the power supply unit for the Interconnect device of the QFabric system.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-power-supply-unit.txt.</p> <p>For more information, see <i>Power Supply Unit MIB</i>.</p> <p>NOTE: On the QFabric system, scalar variables for the jnxPsuObjects 1 object ID in the jnxPsuScalars table are not supported.</p> |

Table 22: Juniper Networks Enterprise-Specific MIBs Supported on QFabric Systems (*continued*)

| MIB | Description |
|----------------------------|---|
| QFabric MIB (jnx-qf-smi) | <p>Explains how the Juniper Networks enterprise-specific QFabric MIBs are structured. Defines the MIB objects that are reported by the QFabric system and the contents of the traps that can be issued by the QFabric system.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-qf-smi.txt.</p> |
| Utility MIB (mib-jnx-util) | <p>Provides you with SNMP MIB container objects of the following types: 32-bit counters, 64-bit counters, signed integers, unsigned integers, and octet strings. You can use these objects to store data that can be retrieved using other SNMP operations.</p> <p>For a downloadable version of this MIB, see http://www.juniper.net/techpubs/en_US/junos13.2/topics/reference/mibs/mib-jnx-util.txt.</p> <p>For more information, see “Utility MIB” on page 76 and “Using the Enterprise-Specific Utility MIB to Enhance SNMP Coverage” on page 176.</p> |

Related Documentation

- *SNMP MIBs and Traps Reference*
- [Understanding the Implementation of SNMP on page 74](#)
- *Understanding the Implementation of SNMP on the QFabric System*
- [SNMP Traps Support on page 100](#)

SNMP Traps Support

The QFX Series standalone switches, QFX Series Virtual Chassis, and QFabric systems support standard SNMP traps and Juniper Networks enterprise-specific traps.

For more information, see:

- [SNMP Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis on page 100](#)
- [SNMP Traps Supported on QFabric Systems on page 109](#)

SNMP Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

QFX Series standalone switches and QFX Series Virtual Chassis support SNMPv1 and v2 traps. For more information, see:

- [SNMPv1 Traps on page 100](#)
- [SNMPv2 Traps on page 105](#)

SNMPv1 Traps

QFX Series standalone switches and QFX Series Virtual Chassis support both standard SNMPv1 traps and Juniper Networks enterprise-specific SNMPv1 traps. See:

- [Table 23 on page 101](#) for standard SNMPv1 traps.
- [Table 24 on page 103](#) for enterprise-specific SNMPv1 traps.

The traps are organized first by trap category and then by trap name. The system logging severity levels are listed for those traps that have them. Traps that do not have corresponding system logging severity levels are marked with an en dash (–).

Table 23: Standard SNMP Version 1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | Syslog Tag |
|--|-------------------------|------------------|---------------------|----------------------|-------------------------------|--------------------------------------|
| Link Notifications | | | | | | |
| RFC 1215, <i>Conventions for Defining Traps for Use with the SNMP</i> | linkDown | 1.3.6.1.4.1.2636 | 2 | 0 | Warning | SNMP_TRAP_LINK_DOWN |
| | linkUp | 1.3.6.1.4.1.2636 | 3 | 0 | Info | SNMP_TRAP_LINK_UP |
| Remote Operations Notifications | | | | | | |
| RFC 2925, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i> | pingProbeFailed | 1.3.6.1.2.1.80.0 | 6 | 1 | Info | SNMP_TRAP_PING_PROBE_FAILED |
| | pingTestFailed | 1.3.6.1.2.1.80.0 | 6 | 2 | Info | SNMP_TRAP_PING_TEST_FAILED |
| | pingTestCompleted | 1.3.6.1.2.1.80.0 | 6 | 3 | Info | SNMP_TRAP_PING_TEST_COMPLETED |
| | traceRoutePathChange | 1.3.6.1.2.1.81.0 | 6 | 1 | Info | SNMP_TRAP_TRACE_ROUTE_PATH_CHANGE |
| | traceRouteTestFailed | 1.3.6.1.2.1.81.0 | 6 | 2 | Info | SNMP_TRAP_TRACE_ROUTE_TEST_FAILED |
| | traceRouteTestCompleted | 1.3.6.1.2.1.81.0 | 6 | 3 | Info | SNMP_TRAP_TRACE_ROUTE_TEST_COMPLETED |
| RMON Alarms | | | | | | |
| RFC 2819a, <i>RMON MIB</i> | fallingAlarm | 1.3.6.1.2.1.16 | 6 | 2 | – | – |
| | risingAlarm | 1.3.6.1.2.1.16 | 6 | 1 | – | – |
| Routing Notifications | | | | | | |

Table 23: Standard SNMP Version 1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | Syslog Tag |
|--|------------------------|---------------------|---------------------|----------------------|-------------------------------|------------------------|
| <i>BGP 4 MIB</i> | bgpEstablished | 1.3.6.1.2.1.15.7 | 6 | 1 | – | – |
| | bgpBackwardTransition | 1.3.6.1.2.1.15.7 | 6 | 2 | – | – |
| <i>OSPF TRAP MIB</i> | ospfVirtIfStateChange | 1.3.6.1.2.1.14.16.2 | 6 | 1 | – | – |
| | ospfNbrStateChange | 1.3.6.1.2.1.14.16.2 | 6 | 2 | – | – |
| | ospfVirtNbrStateChange | 1.3.6.1.2.1.14.16.2 | 6 | 3 | – | – |
| | ospfIfConfigError | 1.3.6.1.2.1.14.16.2 | 6 | 4 | – | – |
| | ospfVirtIfConfigError | 1.3.6.1.2.1.14.16.2 | 6 | 5 | – | – |
| | ospfIfAuthFailure | 1.3.6.1.2.1.14.16.2 | 6 | 6 | – | – |
| | ospfVirtIfAuthFailure | 1.3.6.1.2.1.14.16.2 | 6 | 7 | – | – |
| | ospfIfRxBadPacket | 1.3.6.1.2.1.14.16.2 | 6 | 8 | – | – |
| | ospfVirtIfRxBadPacket | 1.3.6.1.2.1.14.16.2 | 6 | 9 | – | – |
| | ospfTxRetransmit | 1.3.6.1.2.1.14.16.2 | 6 | 10 | – | – |
| | ospfVirtIfTxRetransmit | 1.3.6.1.2.1.14.16.2 | 6 | 11 | – | – |
| | ospfMaxAgeLsa | 1.3.6.1.2.1.14.16.2 | 6 | 13 | – | – |
| | ospfIfStateChange | 1.3.6.1.2.1.14.16.2 | 6 | 16 | – | – |
| Startup Notifications | | | | | | |
| RFC 1215, <i>Conventions for Defining Traps for Use with the SNMP</i> | authenticationFailure | 1.3.6.1.4.1.2636 | 4 | 0 | Notice | SNMPD_TRAP_GEN_FAILURE |
| | coldStart | 1.3.6.1.4.1.2636 | 0 | 0 | Critical | SNMPD_TRAP_COLD_START |
| | warmStart | 1.3.6.1.4.1.2636 | 1 | 0 | Error | SNMPD_TRAP_WARM_START |
| VRRP Notifications | | | | | | |

Table 23: Standard SNMP Version 1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | Syslog Tag |
|--|---------------------|----------------|---------------------|----------------------|-------------------------------|-------------------------|
| RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> | vrrpTrapNewMaster | 1.3.6.1.2.1.68 | 6 | 1 | Warning | VRRPD_NEW_MASTER_TRAP |
| | vrrpTrapAuthFailure | 1.3.6.1.2.1.68 | 6 | 2 | Warning | VRRPD_AUTH_FAILURE_TRAP |

Table 24: Enterprise-Specific SNMPv1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | System Log Tag |
|---|-----------------------|----------------------|---------------------|----------------------|-------------------------------|--------------------|
| Chassis Notifications (Alarm Conditions) | | | | | | |
| <i>Chassis MIB</i> (jnx-chassis.mib) | jnxPowerSupplyFailure | 1.3.6.1.4.1.2636.4.1 | 6 | 1 | Warning | CHASSISD_SNMP_TRAP |
| | jnxFanFailure | 1.3.6.1.4.1.2636.4.1 | 6 | 2 | Critical | CHASSISD_SNMP_TRAP |
| | jnxOverTemperature | 1.3.6.1.4.1.2636.4.1 | 6 | 3 | Alert | CHASSISD_SNMP_TRAP |
| | jnxFruRemoval | 1.3.6.1.4.1.2636.4.1 | 6 | 5 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruInsertion | 1.3.6.1.4.1.2636.4.1 | 6 | 6 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruPowerOff | 1.3.6.1.4.1.2636.4.1 | 6 | 7 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruPowerOn | 1.3.6.1.4.1.2636.4.1 | 6 | 8 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruFailed | 1.3.6.1.4.1.2636.4.1 | 6 | 9 | Warning | CHASSISD_SNMP_TRAP |
| | jnxFruOffline | 1.3.6.1.4.1.2636.4.1 | 6 | 10 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruOnline | 1.3.6.1.4.1.2636.4.1 | 6 | 11 | Notice | CHASSISD_SNMP_TRAP |

Table 24: Enterprise-Specific SNMPv1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | System Log Tag |
|---|--|----------------------|---------------------|----------------------|-------------------------------|--------------------|
| | jnxFruCheck | 1.3.6.1.4.1.2636.4.1 | 6 | 12 | Warning | CHASSISD_SNMP_TRAP |
| | jnxPowerSupplyOk | 1.3.6.1.4.1.2636.4.2 | 6 | 1 | Critical | CHASSISD_SNMP_TRAP |
| | jnxFanOK | 1.3.6.1.4.1.2636.4.2 | 6 | 2 | Critical | CHASSISD_SNMP_TRAP |
| | jnxTemperatureOK | 1.3.6.1.4.1.2636.4.2 | 6 | 3 | Alert | CHASSISD_SNMP_TRAP |
| Configuration Notifications | | | | | | |
| <i>Configuration Management MIB</i> (jnx-configmgmt.mib) | jnxCmCfgChange | 1.3.6.1.4.1.2636.4.5 | 6 | 1 | — | — |
| | jnxCmRescueChange | 1.3.6.1.4.1.2636.4.5 | 6 | 2 | — | — |
| Remote Operations | | | | | | |
| <i>Ping MIB</i> (jnx-ping.mib) | jnxPingRttThresholdExceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 1 | — | — |
| | jnxPingRttStdDevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 2 | — | — |
| | jnxPingRttJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 3 | — | — |
| | jnxPingEgressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 4 | — | — |
| | jnxPingEgressStdDev ThresholdExceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 5 | — | — |
| | jnxPingEgressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 6 | — | — |
| | jnxPingIngressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 7 | — | — |
| | jnxPingIngressStddevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 8 | — | — |
| | jnxPingIngressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9 | 6 | 9 | — | — |

Table 24: Enterprise-Specific SNMPv1 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | Enterprise ID | Generic Trap Number | Specific Trap Number | System Logging Severity Level | System Log Tag |
|------------------------------------|------------------------|----------------------|---------------------|----------------------|-------------------------------|----------------|
| RMON Alarms | | | | | | |
| <i>RMON MIB</i> (jnx-rmon. mib) | jnxRmonAlarmGetFailure | 1.3.6.1.4.1.2636.4.3 | 6 | 1 | – | – |
| | jnxRmonGetOk | 1.3.6.1.4.1.2636.4.3 | 6 | 2 | – | – |

SNMPv2 Traps

- [Table 25 on page 105](#) lists the standard SNMP traps
- [Table 26 on page 107](#) lists the Juniper Networks enterprise-specific traps

Table 25: Standard SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| Defined in | Trap Name | SNMP Trap OID | System Logging Severity Level | Syslog Tag |
|--|-------------------------|---------------------|-------------------------------|--------------------------------------|
| Link Notifications | | | | |
| RFC 2863, <i>The Interfaces Group MIB</i> | linkDown | 1.3.6.1.6.3.1.1.5.3 | Warning | SNMP_TRAP_LINK_DOWN |
| | linkUp | 1.3.6.1.6.3.1.1.5.4 | Info | SNMP_TRAP_LINK_UP |
| Remote Operations Notifications | | | | |
| RFC 2925, <i>Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations</i> | pingProbeFailed | 1.3.6.1.2.1.80.0.1 | Info | SNMP_TRAP_PING_PROBE_FAILED |
| | pingTestFailed | 1.3.6.1.2.1.80.0.2 | Info | SNMP_TRAP_PING_TEST_FAILED |
| | pingTestCompleted | 1.3.6.1.2.1.80.0.3 | Info | SNMP_TRAP_PING_TEST_COMPLETED |
| | traceRoutePathChange | 1.3.6.1.2.1.81.0.1 | Info | SNMP_TRAP_TRACE_ROUTE_PATH_CHANGE |
| | traceRouteTestFailed | 1.3.6.1.2.1.81.0.2 | Info | SNMP_TRAP_TRACE_ROUTE_TEST_FAILED |
| | traceRouteTestCompleted | 1.3.6.1.2.1.81.0.3 | Info | SNMP_TRAP_TRACE_ROUTE_TEST_COMPLETED |

Table 25: Standard SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | SNMP Trap OID | System Logging Severity Level | Syslog Tag |
|---|------------------------|------------------------|-------------------------------|-----------------------|
| RMON Alarms | | | | |
| RFC 2819a, <i>RMON MIB</i> | fallingAlarm | 1.3.6.1.2.1.16.0.1 | – | – |
| | risingAlarm | 1.3.6.1.2.1.16.0.2 | – | – |
| Routing Notifications | | | | |
| <i>BGP 4 MIB</i> | bgpEstablished | 1.3.6.1.2.1.15.7.1 | – | – |
| | bgpBackwardTransition | 1.3.6.1.2.1.15.7.2 | – | – |
| <i>OSPF Trap MIB</i> | ospfVirtIfStateChange | 1.3.6.1.2.1.14.16.2.1 | – | – |
| | ospfNbrStateChange | 1.3.6.1.2.1.14.16.2.2 | – | – |
| | ospfVirtNbrStateChange | 1.3.6.1.2.1.14.16.2.3 | – | – |
| | ospfIfConfigError | 1.3.6.1.2.1.14.16.2.4 | – | – |
| | ospfVirtIfConfigError | 1.3.6.1.2.1.14.16.2.5 | – | – |
| | ospfIfAuthFailure | 1.3.6.1.2.1.14.16.2.6 | – | – |
| | ospfVirtIfAuthFailure | 1.3.6.1.2.1.14.16.2.7 | – | – |
| | ospfIfRxBadPacket | 1.3.6.1.2.1.14.16.2.8 | – | – |
| | ospfVirtIfRxBadPacket | 1.3.6.1.2.1.14.16.2.9 | – | – |
| | ospfTxRetransmit | 1.3.6.1.2.1.14.16.2.10 | – | – |
| | ospfVirtIfTxRetransmit | 1.3.6.1.2.1.14.16.2.11 | – | – |
| | ospfMaxAgeLsa | 1.3.6.1.2.1.14.16.2.13 | – | – |
| | ospfIfStateChange | 1.3.6.1.2.1.14.16.2.16 | – | – |
| Startup Notifications | | | | |
| RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | coldStart | 1.3.6.1.6.3.1.1.5.1 | Critical | SNMPD_TRAP_COLD_START |
| | warmStart | 1.3.6.1.6.3.1.1.5.2 | Error | SNMPD_TRAP_WARM_START |

Table 25: Standard SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Defined in | Trap Name | SNMP Trap OID | System Logging Severity Level | Syslog Tag |
|--|-----------------------|---------------------|-------------------------------|-------------------------|
| | authenticationFailure | 1.3.6.1.6.3.1.1.5.5 | Notice | SNMPD_TRAP_GEN_FAILURE |
| VRRP Notifications | | | | |
| RFC 2787, <i>Definitions of Managed Objects for the Virtual Router Redundancy Protocol</i> | vrrpTrapNewMaster | 1.3.6.1.2.1.68.0.1 | Warning | VRRPD_NEWMASTER_TRAP |
| | vrrpTrapAuthFailure | 1.3.6.1.2.1.68.0.2 | Warning | VRRPD_AUTH_FAILURE_TRAP |

Table 26: Enterprise-Specific SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis

| Source MIB | Trap Name | SNMP Trap OID | System Logging Severity Level | System Log Tag |
|---|-----------------------|-------------------------|-------------------------------|--------------------|
| Chassis (Alarm Conditions) Notifications | | | | |
| <i>Chassis MIB</i> (mib-jnx-chassis) | jnxPowerSupplyFailure | 1.3.6.1.4.1.2636.4.1.1 | Alert | CHASSISD_SNMP_TRAP |
| | jnxFanFailure | 1.3.6.1.4.1.2636.4.1.2 | Critical | CHASSISD_SNMP_TRAP |
| | jnxOverTemperature | 1.3.6.1.4.1.2636.4.1.3 | Critical | CHASSISD_SNMP_TRAP |
| | jnxFruRemoval | 1.3.6.1.4.1.2636.4.1.5 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruInsertion | 1.3.6.1.4.1.2636.4.1.6 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruPowerOff | 1.3.6.1.4.1.2636.4.1.7 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruPowerOn | 1.3.6.1.4.1.2636.4.1.8 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruFailed | 1.3.6.1.4.1.2636.4.1.9 | Warning | CHASSISD_SNMP_TRAP |
| | jnxFruOffline | 1.3.6.1.4.1.2636.4.1.10 | Notice | CHASSISD_SNMP_TRAP |

Table 26: Enterprise-Specific SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Source MIB | Trap Name | SNMP Trap OID | System Logging Severity Level | System Log Tag |
|--|-------------------|--------------------------|-------------------------------|--------------------|
| | jnxFruOnline | 1.3.6.1.4.1.2636.4.1.11 | Notice | CHASSISD_SNMP_TRAP |
| | jnxFruCheck | 1.3.6.1.4.1.2636.4.1.12 | Notice | CHASSISD_SNMP_TRAP |
| | jnxPowerSupplyOK | 1.3.6.1.4.1.2636.4.2.1 | Critical | CHASSISD_SNMP_TRAP |
| | jnxFanOK | 1.3.6.1.4.1.2636.4.2.2 | Critical | CHASSISD_SNMP_TRAP |
| | jnxTemperatureOK | 1.3.6.1.4.1.2636.4.2.3 | Alert | CHASSISD_SNMP_TRAP |
| Configuration Notifications | | | | |
| <i>Configuration Management MIB</i> (mib-jnx-cfgmgmt) | jnxCmCfgChange | 1.3.6.1.4.1.2636.4.5.0.1 | – | – |
| | jnxCmRescueChange | 1.3.6.1.4.1.2636.4.5.0.2 | – | – |
| Remote Operations Notifications | | | | |

Table 26: Enterprise-Specific SNMPv2 Traps Supported on QFX Series Standalone Switches and QFX Series Virtual Chassis (*continued*)

| Source MIB | Trap Name | SNMP Trap OID | System Logging Severity Level | System Log Tag |
|-----------------------------------|--|--------------------------|-------------------------------|----------------|
| <i>Ping MIB</i> (mib-jnx-ping) | jnxPingRttThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.1 | – | – |
| | jnxPingRttStdDevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.2 | – | – |
| | jnxPingRttJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.3 | – | – |
| | jnxPingEgressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.4 | – | – |
| | jnxPingEgressStdDevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.5 | – | – |
| | jnxPingEgressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.6 | – | – |
| | jnxPingIngressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.7 | – | – |
| | jnxPingIngressStddevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.8 | – | – |
| | jnxPingIngressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.9 | – | – |
| RMON Alarms | | | | |
| <i>RMON MIB</i> (mib-jnx-rmon) | jnxRmonAlarmGetFailure | 1.3.6.1.4.1.2636.4.3.0.1 | – | – |
| | jnxRmonGetOk | 1.3.6.1.4.1.2636.4.3.0.2 | – | – |

SNMP Traps Supported on QFabric Systems

QFabric systems support standard SNMPv2 traps and Juniper Networks enterprise-specific SNMPv2 traps.



NOTE: QFabric systems do not support SNMPv1 traps.

For more information, see:

- [Table 27 on page 110](#) for standard SNMPv2 traps

- [Table 28 on page 111](#) for Juniper Networks enterprise-specific SNMPv2 traps

Table 27: Standard SNMPv2 Traps Supported on QFabric Systems

| Defined in | Trap Name | SNMP Trap OID | System Logging Severity Level | Syslog Tag |
|---|-----------------------|---------------------|-------------------------------|------------------------|
| Link Notifications | | | | |
| RFC 2863, <i>The Interfaces Group MIB</i> | linkDown | 1.3.6.1.6.3.1.1.5.3 | Warning | SNMP_TRAP_LINK_DOWN |
| | linkUp | 1.3.6.1.6.3.1.1.5.4 | Info | SNMP_TRAP_LINK_UP |
| Startup Notifications | | | | |
| RFC 1907, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> | coldStart | 1.3.6.1.6.3.1.1.5.1 | Critical | SNMPD_TRAP_COLD_START |
| | warmStart | 1.3.6.1.6.3.1.1.5.2 | Error | SNMPD_TRAP_WARM_START |
| | authenticationFailure | 1.3.6.1.6.3.1.1.5.5 | Notice | SNMPD_TRAP_GEN_FAILURE |

Table 28: Enterprise-Specific SNMPv2 Traps Supported on QFabric Systems

| Source MIB | Trap Name | SNMP Trap OID | System Logging Severity Level | System Log Tag |
|---|--|--------------------------|-------------------------------|----------------|
| <i>Fabric Chassis MIB</i> (mib-jnx-fabric-chassis) | Fabric Chassis (Alarm Conditions) Notifications | | | |
| | jnxFabricPowerSupplyFailure | 1.3.6.1.4.1.2636.4.19.1 | Warning | – |
| | jnxFabricFanFailure | 1.3.6.1.4.1.2636.4.19.2 | Critical | – |
| | jnxFabricOverTemperature | 1.3.6.1.4.1.2636.4.19.3 | Alert | – |
| | jnxFabricRedundancySwitchover | 1.3.6.1.4.1.2636.4.19.4 | Notice | – |
| | jnxFabricFruRemoval | 1.3.6.1.4.1.2636.4.19.5 | Notice | – |
| | jnxFabricFruInsertion | 1.3.6.1.4.1.2636.4.19.6 | Notice | – |
| | jnxFabricFruPowerOff | 1.3.6.1.4.1.2636.4.19.7 | Notice | – |
| | jnxFabricFruPowerOn | 1.3.6.1.4.1.2636.4.19.8 | Notice | – |
| | jnxFabricFruFailed | 1.3.6.1.4.1.2636.4.19.9 | Warning | – |
| | jnxFabricFruOffline | 1.3.6.1.4.1.2636.4.19.10 | Notice | – |
| | jnxFabricFruOnline | 1.3.6.1.4.1.2636.4.19.11 | Notice | – |
| | jnxFabricFruCheck | 1.3.6.1.4.1.2636.4.19.12 | Warning | – |
| | jnxFabricFEBSwitchover | 1.3.6.1.4.1.2636.4.19.13 | Warning | – |
| | jnxFabricHardDiskFailed | 1.3.6.1.4.1.2636.4.19.14 | Warning | – |
| | jnxFabricHardDiskMissing | 1.3.6.1.4.1.2636.4.19.15 | Warning | – |
| | jnxFabricBootFromBackup | 1.3.6.1.4.1.2636.4.19.16 | Warning | – |
| | Fabric Chassis (Alarm Cleared Conditions) Notifications | | | |
| | jnxFabricPowerSupplyOK | 1.3.6.1.4.1.2636.4.20.1 | Critical | – |
| | jnxFabricFanOK | 1.3.6.1.4.1.2636.4.20.2 | Critical | – |
| | jnxFabricTemperatureOK | 1.3.6.1.4.1.2636.4.20.3 | Alert | – |
| | jnxFabricFruOK | 1.3.6.1.4.1.2636.4.20.4 | – | – |

Table 28: Enterprise-Specific SNMPv2 Traps Supported on QFabric Systems (*continued*)

| Source MIB | Trap Name | SNMP Trap OID | System Logging Severity Level | System Log Tag |
|--|--|-----------------------------|-------------------------------|----------------|
| <i>QFabric MIB</i> (mib-jnx-qf-smi) | QFabric MIB Notifications | | | |
| | jnxQFabricDownloadIssued | 1.3.6.1.4.1.2636.3.42.1.0.1 | – | – |
| | jnxQFabricDownloadFailed | 1.3.6.1.4.1.2636.3.42.1.0.2 | – | – |
| | jnxQFabricDownloadSucceeded | 1.3.6.1.4.1.2636.3.42.1.0.3 | – | – |
| | jnxQFabricUpgradeIssued | 1.3.6.1.4.1.2636.3.42.1.0.4 | – | – |
| | jnxQFabricUpgradeFailed | 1.3.6.1.4.1.2636.3.42.1.0.5 | – | – |
| | jnxQFabricUpgradeSucceeded | 1.3.6.1.4.1.2636.3.42.1.0.6 | – | – |
| Configuration Notifications | | | | |
| <i>Configuration Management MIB</i> (mib-jnx-cfgmgmt) | jnxCmCfgChange | 1.3.6.1.4.1.2636.4.5.0.1 | – | – |
| | jnxCmRescueChange | 1.3.6.1.4.1.2636.4.5.0.2 | – | – |
| Remote Operations Notifications | | | | |
| <i>Ping MIB</i> (mib-jnx-ping) | jnxPingRttThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.1 | – | – |
| | jnxPingRttStdDevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.2 | – | – |
| | jnxPingRttJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.3 | – | – |
| | jnxPingEgressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.4 | – | – |
| | jnxPingEgressStdDevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.5 | – | – |
| | jnxPingEgressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.6 | – | – |
| | jnxPingIngressThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.7 | – | – |
| | jnxPingIngressStddevThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.8 | – | – |
| | jnxPingIngressJitterThreshold Exceeded | 1.3.6.1.4.1.2636.4.9.0.9 | – | – |

- Related Documentation**
- [SNMP MIBs and Traps Reference](#)
 - [Understanding the Implementation of SNMP on page 74](#)
 - [Understanding the Implementation of SNMP on the QFabric System](#)
 - [SNMP MIBs Support on page 84](#)

Configuring SNMP

SNMP is implemented in the Junos OS Software running on the QFX Series and OCX Series products. By default, SNMP is not enabled. To enable SNMP, you must include the SNMP configuration statements at the **[edit]** hierarchy level.

To configure the minimum requirements for SNMP, include the following statements at the **[edit]** hierarchy level of the configuration:

```
[edit]
snmp {
  community public;
}
```

To configure complete SNMP features, include the following statements at the **[edit]** hierarchy level of the configuration:

```
snmp {
  client-list client-list-name {
    ip-addresses;
  }
  community community-name {
    authorization authorization;
    client-list-name client-list-name;
    clients {
      address restrict;
    }
    logical-system logical-system-name {
      routing-instance routing-instance-name {
        clients {
          addresses;
        }
      }
    }
    routing-instance routing-instance-name {
      clients {
        addresses;
      }
    }
  }
  view view-name;
}
contact contact;
description description;
filter-duplicates;
filter-interfaces;
health-monitor {
  falling-threshold integer;
```

```

        interval seconds;
        rising-threshold integer;
    }
    interface [ interface-names ];
    location location;
    name name;
    nonvolatile {
        commit-delay seconds;
    }
    rmon {
        alarm index {
            description description;
            falling-event-index index;
            falling-threshold integer;
            falling-threshold-interval seconds;
            interval seconds;
            request-type;
            rising-event-index index;
            rising-threshold integer;
            sample-type (absolute-value | delta-value);
            startup-alarm (falling-alarm | rising-alarm | rising-or-falling alarm);
            syslog-subtag syslog-subtag;
            variable oid-variable;
        }
        event index {
            community community-name;
            description description;
            type type;
        }
        history history-index {
            bucket-size number;
            interface interface-name;
            interval seconds;
            owner owner-name;
        }
    }
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable> <match
            regular-expression>;
        flag flag;
    }
    trap-group group-name {
        categories {
            category;
        }
        destination-port port-number;
        routing-instance routing-instance-name;
        targets {
            address;
        }
        version (all | v1 | v2);
    }
    trap-options {
        agent-address outgoing-interface;
        source-address address;
    }

```

```

v3 {
  notify name {
    tag tag-name;
    type trap;
  }
  notify-filter profile-name {
    oid object-identifier (include | exclude);
  }
  snmp-community community-index {
    community-name community-name;
    security-name security-name;
    tag tag-name;
  }
  target-address target-address-name {
    address address;
    address-mask address-mask;
    logical-system logical-system;
    port port-number;
    retry-count number;
    routing-instance routing-instance-name;
    tag-list tag-list;
    target-parameters target-parameters-name;
    timeout seconds;
  }
  target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
      message-processing-model (v1 | v2c | V3);
      security-level (authentication | none | privacy);
      security-model (usm | v1 | v2c);
      security-name security-name;
    }
  }
}
usm {
  local-engine {
    user username {
      authentication-sha {
        authentication-password authentication-password;
      }
      authentication-md5 {
        authentication-password authentication-password;
      }
      authentication-none;
      privacy-aes128 {
        privacy-password privacy-password;
      }
      privacy-des {
        privacy-password privacy-password;
      }
      privacy-3des {
        privacy-password privacy-password;
      }
      privacy-none;
    }
  }
}
remote-engine engine-id {

```

```

user username {
  authentication-sha {
    authentication-password authentication-password;
  }
  authentication-md5 {
    authentication-password authentication-password;
  }
  authentication-none;
  privacy-aes128 {
    privacy-password privacy-password;
  }
  privacy-des {
    privacy-password privacy-password;
  }
  privacy-3des {
    privacy-password privacy-password;
  }
  privacy-none {
    privacy-password privacy-password;
  }
}
}
}
vacm {
  access {
    group group-name {
      (default-context-prefix | context-prefix context-prefix) {
        security-model (any | usm | v1 | v2c) {
          security-level (authentication | none | privacy) {
            notify-view view-name;
            read-view view-name;
            write-view view-name;
          }
        }
      }
    }
  }
}
security-to-group {
  security-model (usm | v1 | v2c) {
    security-name security-name {
      group group-name;
    }
  }
}
}
}
view view-name {
  oid object-identifier (include | exclude);
}
}

```

- Related Documentation**
- [Understanding the Implementation of SNMP on page 74](#)
 - *snmp*

Configuring the SNMP Community String

The SNMP community string defines the relationship between an SNMP server system and the client systems. This string acts like a password to control the clients' access to the server. To configure a community string in a Junos OS configuration, include the **community** statement at the **[edit snmp]** hierarchy level:

```
[edit snmp]
community name {
  authorization authorization;
  clients {
    default restrict;
    address restrict;
  }
  view view-name;
}
```

If the community name contains spaces, enclose it in quotation marks (" ").

The default authorization level for a community is **read-only**. To allow **Set** requests within a community, you need to define that community as **authorization read-write**. For **Set** requests, you also need to include the specific MIB objects that are accessible with read-write privileges using the **view** statement. The default view includes all supported MIB objects that are accessible with read-only privileges; no MIB objects are accessible with read-write privileges. For more information about the **view** statement, see [“Configuring MIB Views” on page 120](#).

The **clients** statement lists the IP addresses of the clients (community members) that are allowed to use this community. If no **clients** statement is present, all clients are allowed. For **address**, you must specify an IPv4 address, not a hostname. Include the **default restrict** option to deny access to all SNMP clients for which access is not explicitly granted. We recommend that you always include the **default restrict** option to limit SNMP client access to the local switch.



NOTE: Community names must be unique within each SNMP system.

Related Documentation

- [Configuring SNMP on page 113](#)

Configuring SNMP Trap Groups

Before any SNMP traps can be sent, you must configure a trap group, the categories of traps the group can receive, and the targets (systems) that will receive the traps. To create and name an SNMP trap group, include the **trap-group** statement at the **[edit snmp]** hierarchy level:

```
[edit snmp]
trap-group group-name {
  categories {
```

```
    category;  
  }  
  destination-port port-number;  
  targets {  
    address;  
  }  
  version (all | v1 | v2);  
}
```

The trap group name can be any string and is embedded in the community name field of the trap. To configure your own trap group port, include the **destination-port** statement. The default destination port is port 162.

For each trap group that you define, you must include the **target** statement to define at least one system as the recipient of the SNMP traps in the trap group. Specify the IPv4 address of each recipient and not its hostname.

Specify the types of traps the trap group can receive in the **categories** statement.

A trap group can receive the following categories of traps:

- **authentication**—Authentication failures
- **chassis**—Chassis or environment notifications
- **configuration**—Configuration notifications
- **link**—Link-related notifications such as up-down transitions
- **remote-operations**—Remote operation notifications
- **startup**—System warm and cold starts

The **version** statement allows you to specify the SNMP version of the traps sent to targets of the trap group. If you specify **v1** only, SNMPv1 traps are sent. If you specify **v2** only, SNMPv2 traps are sent. If you specify **all**, both an SNMPv1 and an SNMPv2 trap are sent for every trap condition. For more information about the **version** statement, see *version*.

Adding a Group of Clients to an SNMP Community

Junos OS enables you to add one or more groups of clients to an SNMP community. You can include the **client-list-name** *name* statement at the **[edit snmp community community-name]** hierarchy level to add all the members of the client list or prefix list to an SNMP community.

To define a list of clients, include the **client-list** statement followed by the IP addresses of the clients at the **[edit snmp]** hierarchy level:

```
[edit snmp]  
  client-list client-list-name {  
    ip-addresses;  
  }
```

You can configure a prefix list at the **[edit policy options]** hierarchy level. Support for prefix lists in the SNMP community configuration enables you to use a single list to configure the SNMP and routing policies. For more information about the **prefix-list**

statement, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.

To add a client list or prefix list to an SNMP community, include the **client-list-name** statement at the **[edit snmp community *community-name*]** hierarchy level:

```
[edit snmp community community-name]
client-list-name client-list-name;
```



NOTE: The client list and prefix list must not have the same name.

The following example shows how to define a client list:

```
[edit]
snmp {
  client-list clentlist1 {
    10.1.1.1/32;
    10.2.2.2/32;
  }
}
```

The following example shows how to add a client list to an SNMP community:

```
[edit]
snmp {
  community community1 {
    authorization read-only;
    client-list-name clientlist1;
  }
}
```

The following example shows how to add a prefix list to an SNMP community:

```
[edit]
policy-options {
  prefix-list prefixlist {
    10.3.3.3/32;
    10.5.5.5/32;
  }
}
snmp {
  community community2 {
    client-list-name prefixlist;
  }
}
```

- Related Documentation**
- *client-list*
 - *client-list-name*

Configuring the Interfaces on Which SNMP Requests Can Be Accepted

By default, all router or switch interfaces have SNMP access privileges. To limit the access through certain interfaces only, include the **interface** statement at the **[edit snmp]** hierarchy level:

```
[edit snmp]
interface [ interface-names ];
```

Specify the names of any logical or physical interfaces that should have SNMP access privileges. Any SNMP requests entering the router or switch from interfaces not listed are discarded.

Related Documentation

- *Configuring SNMP on a Device Running Junos OS*
- *Configuration Statements at the [edit snmp] Hierarchy Level*
- *Example: Configuring Secured Access List Checking*
- [Configuring SNMP on page 113](#)

Configuring MIB Views

SNMPv3 defines the concept of MIB views in RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*. MIB views provide an agent better control over who can access specific branches and objects within its MIB tree. A view consists of a name and a collection of SNMP object identifiers, which are either explicitly included or excluded. Once defined, a view is then assigned to an SNMPv3 group or SNMPv1/v2c community (or multiple communities), automatically masking which parts of the agent's MIB tree members of the group or community can (or cannot) access.

By default, an SNMP community grants read access and denies write access to all supported MIB objects (even communities configured as **authorization read-write**). To restrict or grant read or write access to a set of MIB objects, you must configure a MIB view and associate the view with a community.

To configure MIB views, include the **view** statement at the **[edit snmp]** hierarchy level:

```
[edit snmp]
view view-name {
  oid object-identifier (include | exclude);
}
```

The **view** statement defines a MIB view and identifies a group of MIB objects. Each MIB object of a view has a common object identifier (OID) prefix. Each object identifier represents a subtree of the MIB object hierarchy. The subtree can be represented either by a sequence of dotted integers (such as **1.3.6.1.2.1.2**) or by its subtree name (such as **interfaces**). A configuration statement uses a view to specify a group of MIB objects on which to define access. You can also use a wildcard character asterisk (*) to include OIDs that match a particular pattern in the SNMP view. To enable a view, you must associate the view with a community.



NOTE: To remove an OID completely, use the `delete view all oid oid-number` command but omit the `include` parameter.

To associate MIB views with a community, include the **view** statement at the `[edit snmp community community-name]` hierarchy level:

```
[edit snmp community community-name]  
view view-name;
```

For more information about the Ping MIB, see RFC 2925 and the *PING MIB* topic.

Related Documentation

- [PING MIB](#)
- [Configuring SNMP on a Device Running Junos OS](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level](#)
- [Example: Ping Proxy MIB](#)
- [view \(Configuring a MIB View\)](#)
- [view \(Associating MIB View with a Community\)](#)
- [oid](#)

Configuring RMON Alarms and Events

The Junos OS supports the *Remote Network Monitoring* (RMON) MIB (RFC 2819), which allows a management device to monitor the values of MIB objects, or variables, against configured thresholds. When the value of a variable crosses a threshold, an alarm and its corresponding event are generated. The event can be logged and can generate an SNMP trap.

To configure RMON alarms and events using the CLI, perform these tasks:

1. [Configuring SNMP on page 121](#)
2. [Configuring an Event on page 122](#)
3. [Configuring an Alarm on page 122](#)

Configuring SNMP

To configure SNMP:

1. Grant read-only access to all SNMP clients:

```
[edit snmp]  
user@switch# set community community-name authorization authorization  
For example:
```

```
[edit snmp]  
user@switch# set community public authorization read-only
```

2. Grant read-write access to the **RMON** and **jnx-rmon** MIBs:

```
[edit snmp]
```

```
user@switch# set view view-name oid object-identifier include
user@switch# set view view-name oid object-identifier include
user@switch# set community community-name authorization authorization view view-name
```

For example:

```
[edit snmp]
user@switch# set view rmon-mib-view oid .1.3.6.1.2.1.16 include
user@switch# set view rmon-mib-view oid .1.3.6.1.4.1.2636.13 include
user@switch# set community private authorization read-write view rmon-mib-view
```

OIDs 1.3.6.1.2.1.16 and 1.3.6.1.4.1.2636.13 correspond to the **RMON** and **jnxRmon** MIBs.

3. Configure an SNMP trap group:

```
[edit snmp]
user@switch# set trap-group group-name categories category
user@switch# set trap-group group-name targets address
```

For example:

```
[edit snmp]
user@switch# set trap-group rmon-trap-group categories rmon-alarm
user@switch# set trap-group rmon-trap-group targets 192.168.5.5
```

The trap group **rmon-trap-group** is configured to send RMON traps to 192.168.5.5.

Configuring an Event

To configure an event:

1. Configure an event index, community name, and type:

```
[edit snmp rmon]
user@switch# set event index community community-name typetype
```

For example:

```
[edit snmp rmon]
user@switch# set event 1 community rmon-trap-group type log-and-trap
```

The event community corresponds to the SNMP trap group and is not the same as an SNMP community. This event generates an SNMP trap and adds an entry to the **logTable** in the RMON MIB.

2. Configure a description for the event:

```
[edit snmp rmon]
user@switch# set event index description description
```

For example:

```
[edit snmp rmon]
user@switch# set event 1 description "rmon event"
```

Configuring an Alarm

To configure an alarm:

1. Configure an alarm index, the variable to monitor, the rising and falling thresholds, and the corresponding rising and falling events:

```
[edit snmp rmon]
user@switch# set alarm index variable oid-variable falling-threshold integer rising-threshold integer rising-event-index index falling-event-index index
```

For example:

```
[edit snmp rmon]
user@switch# set alarm 5 variable .1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0 falling-threshold 75
rising-threshold 90 rising-event-index 1 falling-event-index 1
```

The variable .1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0 corresponds to the **jnxRmon** MIB object **jnxOperatingCPU**, which represents the CPU utilization of the Routing Engine. The falling and rising threshold integers are 75 and 90. The rising and falling events both generate the same event (event index 1).

2. Configure the sample interval and type and the alarm type:

```
[edit snmp rmon]
user@switch# set alarm index interval seconds sample-type (absolute-value | delta-value)
startup-alarm (falling-alarm | rising-alarm | rising-or-falling-alarm)
```

For example:

```
[edit snmp rmon]
user@switch# set alarm 5 interval 30 sample-type absolute-value
startup-alarm rising-or-falling-alarm
```

The absolute value of the monitored variable is sampled every 30 seconds. The initial alarm can occur because of rising above the rising threshold or falling below the falling threshold.

Related Documentation

- [Configuring SNMP on page 113](#)
- [Juniper Networks Enterprise-Specific MIBs](#)
- [Monitoring RMON MIB Tables on page 169](#)
- [RMON MIB Event, Alarm, Log, and History Control Tables on page 81](#)
- [Understanding RMON on page 79](#)

Configuring Health Monitoring

This topic describes how to configure the health monitor feature for QFX Series and OCX Series devices.

The health monitor feature extends the SNMP RMON alarm infrastructure to provide predefined monitoring for a selected set of object instances (such as file system usage, CPU usage, and memory usage) and dynamic object instances (such as Junos OS processes).

To configure health monitoring:

1. Configure the health monitor:

```
[edit snmp]
user@switch# set health-monitor
```

2. Configure the falling threshold:

```
[edit snmp]
user@switch# set health-monitor falling-threshold percentage
```

For example:

```
user@switch# set health-monitor falling-threshold 85
```

3. Configure the rising threshold:

```
[edit snmp]
user@switch# set health-monitor rising-threshold percentage
```

For example:

```
user@switch# set health-monitor rising-threshold 75
```

4. Configure the interval:

```
[edit snmp]
user@switch# set health-monitor interval seconds
```

For example:

```
user@switch# set health-monitor interval 600
```

- Related Documentation**
- [Understanding Health Monitoring on page 83](#)
 - *falling-threshold*
 - *interval (Health Monitor)*
 - *rising-threshold (Health Monitor)*

Creating SNMPv3 Users

For each SNMPv3 user, you can specify the username, authentication type, authentication password, privacy type, and privacy password. After a user enters a password, a key based on the engine ID and password is generated and is written to the configuration file. After the generation of the key, the password is deleted from this configuration file.



NOTE: You can configure only one encryption type for each SNMPv3 user.

To create users, include the **user** statement at the **[edit snmp v3 usm local-engine]** hierarchy level:

```
[edit snmp v3 usm local-engine]
user username;
```

username is the name that identifies the SNMPv3 user.

To configure user authentication and encryption, include the following statements at the **[edit snmp v3 usm local-engine user *username*]** hierarchy level:

```
[edit snmp v3 usm local-engine user username]
authentication-md5 {
  authentication-password authentication-password;
}
authentication-sha {
  authentication-password authentication-password;
}
authentication-none;
```

```

privacy-aes128 {
    privacy-password privacy-password;
}
privacy-des {
    privacy-password privacy-password;
}
privacy-3des {
    privacy-password privacy-password;
}
privacy-none;

```

**Related
Documentation**

- [Complete SNMPv3 Configuration Statements](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78](#)
- [Example: Creating SNMPv3 Users](#)
- [Example: SNMPv3 Configuration](#)

Configuring Access Privileges for a Group

In SNMPv3, you can configure a group that sets the same access privileges for one or more users. Configuring a group includes defining the security model and security level, and associating one or more MIB view permissions for the group.



NOTE: You must associate at least one MIB view with the group. You can associate multiple MIB views (read, notify, write) to authorize different permissions based on the view. The view name cannot exceed 32 characters.

To configure access privileges for a group:

1. To configure the group:

```

[edit snmp v3 vacm access]
user@switch# edit group group-name

```

2. To configure the context prefix of the SNMP instance for the group:

```

[edit snmp v3 vacm access group group-name]
user@switch# edit (default-context-prefix | context-prefix context-prefix)

```

For example, to configure the default context prefix:

```

[edit snmp v3 vacm access group group-name]
user@switch# edit default-context-prefix

```

3. To configure the security model:

```

[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix context-prefix)]
user@switch# edit security-model (any | usm | v1 | v2c)

```

For example, to configure the SNMPv3 user-based security model (USM):

```

[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix context-prefix)]

```

```
user@switch# edit security-model usm
```

4. To configure the security level:

```
[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix
context-prefix) security-model (any | usm | v1 | v2c)]
user@switch# edit security-level (authentication | none | privacy)
```

For example, to configure a security level requiring user authentication and encryption:

```
[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix
context-prefix) security-model (any | usm | v1 | v2c)]
user@switch# edit security-level privacy
```



NOTE: Access privileges are granted to all packets with a security level equal to or greater than that configured. If you are configuring the SNMPv1 or v2c security model, use *none* as your security level. If you are configuring the SNMPv3 security model (USM), use the *authentication*, *none*, or *privacy* security level.

5. (Optional) To associate a read-only MIB view with an SNMP group:

```
[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix
context-prefix) security-model (any | usm | v1 | v2c) security-level (authentication |
none | privacy)]
user@switch# edit read-view view-name
```

6. (Optional) To associate a MIB view with an SNMP notification permission for an SNMP group:

```
[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix
context-prefix) security-model (any | usm | v1 | v2c) security-level (authentication |
none | privacy)]
user@switch# edit notify-view view-name
```

7. (Optional) To associate a MIB view with write permission for an SNMP group:

```
[edit snmp v3 vacm access group group-name (default-context-prefix | context-prefix
context-prefix) security-model (any | usm | v1 | v2c) security-level (authentication |
none | privacy)]
user@switch# edit write-view view-name
```

- Related Documentation**
- [SNMPv3 Overview on page 77](#)
 - [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78](#)

Assigning a Security Name to a Group

In SNMPv3, each username is associated with a security name. The security name, together with the SNMP engine ID, is included in SNMP messages to ensure messaging security.

Before you assign a security name to a group, first create the security name. For an SNMPv3 client, the security name is the username configured at the `[edit snmp v3 usm`

local-engine user *username*] hierarchy level. For SNMPv1 or v2c clients, the security name is the community string configured at the **[edit snmp v3 snmp-community *community-index*]** hierarchy level.

Assigning a security name to a group includes configuring a security model for the group, assigning the security name to the group, and configuring the group.

To assign an SNMP security name to a group:

1. To configure a security model for the group:

```
[edit snmp v3 vacm security-to-group]
user@switch# edit security-model (usm | v1 | v2c)
```

For example, to configure the SNMPv3 user-based security model (USM):

```
[edit snmp v3 vacm security-to-group]
user@switch# edit security-model usm
```

2. To associate the security name with a group:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c)]
user@switch# edit security-name security-name
```

3. To configure a group of SNMPv3 security names with the same security policy:

```
[edit snmp v3 vacm security-to-group security-model (usm | v1 | v2c) security-name
security-name]
user@switch# edit group group-name
```

Related Documentation

- [Creating SNMPv3 Users on page 124](#)
- *group* (Associating a Security Name)
- *security-model* (Group)
- *security-name* (Community String)
- *security-name* (Security Group)

Configuring SNMPv3 Traps on a Device Running Junos OS

In SNMPv3, you create traps and informs by configuring the **notify**, **target-address**, and **target-parameters** parameters. Traps are unconfirmed notifications, whereas informs are confirmed notifications. This section describes how to configure SNMP traps. For information about configuring SNMP informs, see [“Configuring SNMP Informs” on page 128](#).

The target address defines a management application’s address and parameters to be used in sending notifications. Target parameters define the message processing and security parameters that are used in sending notifications to a particular management target. SNMPv3 also lets you define SNMPv1 and SNMPv2c traps.



NOTE: When you configure SNMP traps, make sure your configured access privileges allow the traps to be sent. Access privileges are configured at the `[edit snmp v3 vacm access]` and `[edit snmp v3 vacm security-to-group]` hierarchy levels.

To configure SNMP traps, include the following statements at the `[edit snmp v3]` hierarchy level:

```
[edit snmp v3]
  notify name {
    tag tag-name;
    type trap;
  }
  notify-filter name {
    oid object-identifier (include | exclude);
  }
  target-address target-address-name {
    address address;
    address-mask address-mask;
    logical-system (SNMP) logical-system;
    port port-number;
    routing-instance instance;
    tag-list tag-list;
    target-parameters target-parameters-name;
  }
  target-parameters target-parameters-name {
    notify-filter profile-name;
    parameters {
      message-processing-model (v1 | v2c | v3);
      security-level (authentication | none | privacy);
      security-model (usm | v1 | v2c);
      security-name security-name;
    }
  }
}
```

Related Documentation

- [Configuring the SNMPv3 Trap Notification](#)
- [Configuring the Trap Notification Filter](#)
- [Configuring the Trap Target Address](#)
- [Defining and Configuring the Trap Target Parameters](#)
- [Configuring SNMP Informs on page 128](#)
- [Configuring the Remote Engine and Remote User](#)
- [Configuring the Inform Notification Type and Target Address](#)

Configuring SNMP Informs

Junos OS supports two types of notifications: traps and informs. With traps, the receiver does not send any acknowledgment when it receives a trap. Therefore, the sender cannot

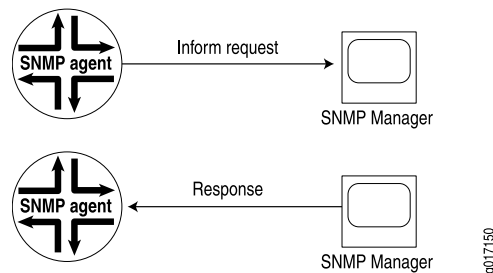
determine if the trap was received. A trap may be lost because a problem occurred during transmission. To increase reliability, an inform is similar to a trap except that the inform is stored and retransmitted at regular intervals until one of these conditions occurs:

- The receiver (target) of the inform returns an acknowledgment to the SNMP agent.
- A specified number of unsuccessful retransmissions have been attempted and the agent discards the inform message.

If the sender never receives a response, the inform can be sent again. Thus, informs are more likely to reach their intended destination than traps are. Informs use the same communications channel as traps (same socket and port) but have different protocol data unit (PDU) types.

Informs are more reliable than traps, but they consume more network, router, and switch resources (see [Figure 8 on page 129](#)). Unlike a trap, an inform is held in memory until a response is received or the timeout is reached. Also, traps are sent only once, whereas an inform may be retried several times. Use informs when it is important that the SNMP manager receive all notifications. However, if you are more concerned about network traffic, or router and switch memory, use traps.

Figure 8: Inform Request and Response



For information about configuring SNMP traps, see [“Configuring SNMPv3 Traps on a Device Running Junos OS” on page 127](#).

Related Documentation

- [Configuring SNMPv3 Traps on a Device Running Junos OS on page 127](#)
- [Configuring the Remote Engine and Remote User](#)
- [Configuring the Inform Notification Type and Target Address](#)
- [Complete SNMPv3 Configuration Statements](#)
- [Minimum SNMPv3 Configuration on a Device Running Junos OS on page 78](#)

CHAPTER 7

Configuring System Logging

- Overview of Junos OS System Log Messages on page 132
- Overview of Single-Chassis System Logging Configuration on page 132
- Examples: Configuring System Logging on page 134
- Examples: Assigning an Alternative Facility on page 136
- Junos OS Minimum System Logging Configuration on page 137
- Junos OS System Log Configuration Statements on page 137
- Adding a Text String to System Log Messages on page 138
- Directing System Log Messages to a Log File on page 139
- Directing System Log Messages to a Remote Machine on page 140
- Directing System Log Messages to a User Terminal on page 140
- Directing System Log Messages to the Console on page 141
- Disabling the System Logging of a Facility on page 141
- Displaying a Log File from a Single-Chassis System on page 142
- Including Priority Information in System Log Messages on page 143
- Including the Year or Millisecond in Timestamps on page 144
- Logging Messages in Structured-Data Format on page 145
- Interpreting Messages Generated in Structured-Data Format on page 146
- Interpreting Messages Generated in Standard Format on page 149
- Specifying Log File Size, Number, and Archiving Properties on page 150
- Specifying the Facility and Severity of Messages to Include in the Log on page 151
- Junos OS System Logging Facilities and Message Severity Levels on page 152
- System Log Default Facilities for Messages Directed to a Remote Destination on page 153
- Alternate Facilities for System Log Messages Directed to a Remote Destination on page 154
- Changing the Alternative Facility Name for System Log Messages Directed to a Remote Destination on page 155
- Using Regular Expressions to Refine the Set of Logged Messages on page 156

Overview of Junos OS System Log Messages

The Junos OS generates system log messages (also called *syslog messages*) to record events that occur on the switch, including the following:

- Routine operations, such as a user login into the configuration database.
- Failure and error conditions, such as failure to access a configuration file.
- Emergency or critical conditions, such as power-down of the switch due to excessive temperature.

Each system log message identifies the Junos OS process that generated the message and briefly describes the operation or error that occurred. For detailed information about specific system log messages, see the *Junos OS System Log Messages Reference*.



NOTE: OCX Series switches comprise both the Junos OS and the host operating system (OS). For information about system logging on the host OS, see *Managing Host OS System Log and Core Files*.

Related Documentation

- [Junos OS System Log Configuration Statements on page 137](#)
- [Junos OS Minimum System Logging Configuration on page 137](#)

Overview of Single-Chassis System Logging Configuration

The Junos OS system logging utility on the QFX Series is similar to the UNIX **syslogd** utility. This topic describes how to configure system logging for a single-chassis system that runs the Junos OS.

Each system log message belongs to a *facility*, which groups together related messages. Each message is also preassigned a *severity level*, which indicates how seriously the triggering event affects router functions. You always specify the facility and severity of the messages to include in the log. For more information, see *Specifying the Facility and Severity of Messages to Include in the Log*.

You direct messages to one or more destinations by including the appropriate statement at the **[edit system syslog]** hierarchy level:

- To a named file in a local file system, by including the **file** statement. See [“Directing System Log Messages to a Log File” on page 139](#).
- To the terminal session of one or more specific users (or all users) when they are logged in to the switch, by including the **user** statement. See [“Directing System Log Messages to a User Terminal” on page 140](#).

- To the switch console, by including the **console** statement. See [“Directing System Log Messages to the Console” on page 141](#).
- To a remote machine that is running the **syslogd** utility, by including the **host** statement. See [“Directing System Log Messages to a Remote Machine” on page 140](#).

By default, messages are logged in a standard format, which is based on a UNIX system log format; for detailed information about message formatting, see the *Junos OS System Log Messages Reference*. You can alter the content and format of logged messages in the following ways:

- You can log messages to a file in structured-data format instead of the standard Junos OS format. Structured-data format provides more information without adding significant length, and makes it easier for automated applications to extract information from the message. For more information, see [“Logging Messages in Structured-Data Format” on page 145](#).
- A message’s facility and severity level are together referred to as its *priority*. By default, the standard Junos OS format for messages does not include priority information (structured-data format includes a priority code by default). To include priority information in standard-format messages directed to a file or a remote destination, include the **explicit-priority** statement. For more information, see [“Including Priority Information in System Log Messages” on page 143](#).
- By default, the standard Junos OS format for messages specifies the month, date, hour, minute, and second when the message was logged. You can modify the timestamp on standard-format system log messages to include the year, the millisecond, or both. (Structured-data format specifies the year and millisecond by default.) For more information, see [“Including the Year or Millisecond in Timestamps” on page 144](#).
- When directing messages to a remote machine, you can specify the IP address that is reported in messages as their source. You can also configure features that make it easier to separate messages generated by Junos OS or messages generated on particular switches. For more information, see [“Directing System Log Messages to a Remote Machine” on page 140](#).
- The predefined facilities group together related messages, but you can also use regular expressions to specify more exactly which messages from a facility are logged to a file, a user terminal, or a remote destination. For more information, see [“Using Regular Expressions to Refine the Set of Logged Messages” on page 156](#).



NOTE: During a commit check, warnings about the **traceoptions** configuration (for example, mismatch in trace file sizes or number of trace files) are not displayed on the console. However, these warnings are logged in the system log messages when the new configuration is committed.

Related Documentation

- [Examples: Configuring System Logging on page 134](#)
- [Specifying the Facility and Severity of Messages to Include in the Log](#)
- [Junos OS System Logging Facilities and Message Severity Levels on page 152](#)

- [Directing System Log Messages to a Log File on page 139](#)
- [Directing System Log Messages to a Remote Machine on page 140](#)
- [Directing System Log Messages to a User Terminal on page 140](#)
- [Directing System Log Messages to the Console on page 141](#)

Examples: Configuring System Logging

The system log provides an excellent way of tracking all management activity on the switch by recording events such as user authentication, access authorization, and command execution. Logged command executions include commands entered by users at the CLI prompt or by client applications such as the Junos XML protocol or NETCONF XML client. Because system log files contain information about commands executed on the switch and the user who executed the commands, checking system log files for failed authentication events can help identify attempts to hack in to the switch. You can also analyze network activity by correlating executed commands with events and changes that occurred on the network at a particular time.

System log files are stored locally on the switch in the default `/var/log` directory.

The following example shows how to configure system log messages to record all commands entered by users and all authentication or authorization attempts. Logged commands include those entered by users at the CLI prompt and by client applications. Authentication and authorization attempts include events that are saved in the file named **cli-commands** and those that are sent to the terminal of a user who is logged in.

```
[edit system]
syslog {
  file cli-commands {
    interactive-commands info;
    authorization info;
  }
  user * {
    interactive-commands info;
    authorization info;
  }
}
```

The following example shows how to log all alarms state changes to the file `/var/log/alarms`:

```
[edit system]
syslog {
  file alarms {
    kernel warning;
  }
}
```

The following example shows how to configure the handling of messages of various types, as described in the comments. Information is logged to two files, to the terminal of user alex, to a remote machine, and to the console:

```

[edit system]
syslog {
  /* write all security-related messages to file /var/log/security */
  file security {
    authorization info;
    interactive-commands info;
  }
  /* write messages about potential problems to file /var/log/messages: */
  /* messages from "authorization" facility at level "notice"
  and above, */
  /* messages from all other facilities at level "warning" and above */
  file messages {
    authorization notice;
    any warning;
  }
  /* write all messages at level "critical" and above to terminal of user
  "alex" if */
  /* that user is logged in */
  user alex {
    any critical;
  }
  /* write all messages from the "daemon" facility at level "info"
  and above, and */
  /* messages from all other facilities at level "warning" and above, to the
  */
  /* machine monitor.mycompany.com */
  host monitor.mycompany.com {
    daemon info;
    any warning;
  }
  /* write all messages at level "error" and above to the system console */
  console {
    any error;
  }
}

```

The following example shows how to configure the handling of messages generated when users issue Junos OS CLI commands, by specifying the interactive-commands facility at the info, notice, and warning severity levels:

```

[edit system]
file user-actions {
  interactive-commands info;
}
user philip {
  interactive-commands notice;
}
console {
  interactive-commands warning;
}

```

The following list describes the security levels used in the example:

- **info**—Logs a message when users issue any command at the CLI operational or configuration mode prompt. The example writes the messages to the file `/var/log/user-actions`.
- **notice**—Logs a message when users issue the configuration mode command **commit**. The example writes the messages to the terminal of user philip.
- **warning**—Logs a message when users issue a command that restarts a software process. The example writes the messages to the console.

**Related
Documentation**

- [Overview of Single-Chassis System Logging Configuration on page 132](#)

Examples: Assigning an Alternative Facility

This topic contains examples of configuring system log messages to use an alternative facility for logging.

The following example shows how to log all messages generated on the switch at the **error** level or higher to the **local0** facility on the remote host called **monitor.mycompany.com**:

```
[edit system syslog]
host monitor.mycompany.com {
    any error;
    facility-override local0;
}
```

The following example contains two sets of statements that show how to configure switches located in California and in New York to send messages to a single remote host called **central-logger.mycompany.com**. The messages from California are assigned to alternative facility **local0** and the messages from New York are assigned to alternative facility **local2**.

- The following statements configure the California switch to aggregate messages in the **local0** facility:

```
[edit system syslog]
host central-logger.mycompany.com {
    change-log info;
    facility-override local0;
}
```

- The following statements configure the New York switch to aggregate messages in the **local2** facility:

```
[edit system syslog]
host central-logger.mycompany.com {
    change-log info;
    facility-override local2;
}
```


On the remote host named **central-logger** you can subsequently configure the system logging utility to write messages from the **local0** facility to one file (for example, **california-config**) and the messages from the **local2** facility to another file (for example, **new-york-config**).

- Related Documentation
- [Alternate Facilities for System Log Messages Directed to a Remote Destination on page 154](#)

Junos OS Minimum System Logging Configuration

To record or view system log messages, you must include the **syslog** statement at the **[edit system]** hierarchy level. Specify at least one destination for the messages, as described in [Table 29 on page 137](#). For more information about the configuration statements, see *Single-Chassis System Logging Configuration Overview*.

Table 29: Minimum Configuration Statements for System Logging

| Destination | Minimum Configuration Statements |
|--|---|
| File | <pre>[edit system syslog] file filename { facility severity; }</pre> |
| Terminal session of one, several, or all users | <pre>[edit system syslog] user (username *) { facility severity; }</pre> |
| Router or switch console | <pre>[edit system syslog] console { facility severity; }</pre> |
| Remote machine or the other Routing Engine on the router or switch | <pre>[edit system syslog] host (hostname other-routing-engine) { facility severity; }</pre> |

- Related Documentation
- [Junos OS System Log Overview](#)
 - [Overview of Junos OS System Log Messages on page 132](#)
 - [Overview of Single-Chassis System Logging Configuration on page 132](#)

Junos OS System Log Configuration Statements

To configure the switch to log system messages, include the **syslog** statement at the **[edit system]** hierarchy level:

```
[edit system]
syslog {
  archive <files number> <size size> <world-readable | no-world-readable>;
```

```
console {  
    facility severity;  
}  
file filename {  
    facility severity;  
    archive <archive-sites (ftp-url <password password>) <files number> <size size>  
        <start-time "YYYY-MM-DD.hh:mm"> <transfer-interval minutes> <world-readable |  
        no-world-readable>;  
    explicit-priority;  
    match "regular-expression";  
    structured-data {  
        brief;  
    }  
}  
host hostname {  
    facility severity;  
    explicit-priority;  
    facility-override facility;  
    log-prefix string  
    match "regular-expression";  
}  
source-address source-address;  
time-format (year | millisecond | year millisecond);  
user (username | *) {  
    facility severity;  
    match "regular-expression";  
}  
}
```

Related Documentation • [Overview of Junos OS System Log Messages on page 132](#)

Adding a Text String to System Log Messages

To add a text string to every system log message directed to a remote machine or to the other Routing Engine, include the **log-prefix** statement at the **[edit system syslog host]** hierarchy level:

```
[edit system syslog host (hostname | other-routing-engine)]  
    facility severity;  
    log-prefix string;
```

The string can contain any alphanumeric or special character except the equal sign (=) and the colon (:). It also cannot include the space character; do not enclose the string in quotation marks (" ") in an attempt to include spaces in it.

The Junos OS system logging utility automatically appends a colon and a space to the specified string when the system log messages are written to the log. The string is inserted after the identifier for the Routing Engine that generated the message.

The following example shows how to add the string M120 to all messages to indicate that the router is an M120 router, and direct the messages to the remote machine hardware-logger.mycompany.com:

```
[edit system syslog]
```

```

host hardware-logger.mycompany.com {
    any info;
    log-prefix M120;
}

```

When these configuration statements are included on an M120 router called origin1, a message in the system log on hardware-logger.mycompany.com looks like the following:

```

Mar 9 17:33:23 origin1 M120:mgd[477]: UI_CMDLINE_READ_LINE: user 'root', command 'run
show version'

```

**Related
Documentation**

- [Single-Chassis System Logging Configuration Overview](#)
- [Specifying Log File Size, Number, and Archiving Properties on page 150](#)

Directing System Log Messages to a Log File

To direct system log messages to a file in the `/var/log` directory of the local Routing Engine, include the `file` statement at the `[edit system syslog]` hierarchy level:

```

[edit system syslog]
file filename {
    facility severity;
    archive <archive-sites (ftp-url <password password>)> <files number> <size size>
        <start-time "YYYY-MM-DD.hh:mm"> <transfer-interval minutes> <world-readable |
        no-world-readable>;
    explicit-priority;
    match "regular-expression";
    structured-data {
        brief;
    }
}

```

For the list of facilities and severity levels, see *Specifying the Facility and Severity of Messages to Include in the Log*.

To prevent log files from growing too large, the Junos OS system logging utility by default writes messages to a sequence of files of a defined size. By including the **archive** statement, you can configure the number of files, their maximum size, and who can read them, either for all log files or for a certain log file. For more information, see [“Specifying Log File Size, Number, and Archiving Properties” on page 150](#).

For information about the following statements, see the indicated sections:

- **explicit-priority**—See [“Including Priority Information in System Log Messages” on page 143](#)
- **match**—See [“Using Regular Expressions to Refine the Set of Logged Messages” on page 156](#)
- **structured-data**—See *Logging Messages in Structured-Data Format*

**Related
Documentation**

- [Single-Chassis System Logging Configuration Overview](#)
- [Overview of Junos OS System Log Messages on page 132](#)

- [Logging Messages in Structured-Data Format](#)
- [Examples: Configuring System Logging](#)

Directing System Log Messages to a Remote Machine

To direct system log messages to a remote machine, include the **host** statement at the **[edit system syslog]** hierarchy level:

```
[edit system syslog]
host (hostname | other-routing-engine) {
    facility severity;
    explicit-priority;
    facility-override facility;
    log-prefix string;
    match "regular-expression";
}
source-address source-address;
```

To direct system log messages to a remote machine, include the **host hostname** statement to specify the remote machine's IP version 4 (IPv4) address or fully qualified hostname. The remote machine must be running the standard **syslogd** utility. We do not recommend directing messages to another Juniper Networks switch. In each system log message directed to the remote machine, the hostname of the local Routing Engine appears after the timestamp to indicate that it is the source for the message.

For the list of logging facilities and severity levels to configure under the **host** statement, see [Specifying the Facility and Severity of Messages to Include in the Log](#).

To record facility and severity level information in each message, include the **explicit-priority** statement. For more information, see ["Including Priority Information in System Log Messages" on page 143](#).

For information about the **match** statement, see ["Using Regular Expressions to Refine the Set of Logged Messages" on page 156](#).

When directing messages to remote machines, you can include the **source-address** statement to specify the IP address of the switch that is reported in the messages as their source. In each **host** statement, you can also include the **facility-override** statement to assign an alternative facility and the **log-prefix** statement to add a string to each message.

Related Documentation

- [Overview of Single-Chassis System Logging Configuration on page 132](#)

Directing System Log Messages to a User Terminal

To direct system log messages to the terminal session of one or more specific users (or all users) when they are logged in to the local Routing Engine, include the **user** statement at the **[edit system syslog]** hierarchy level:

```
[edit system syslog]
```

```

user (username | *) {
    facility severity;
    match "regular-expression";
}

```

Specify one or more Junos OS usernames, separating multiple values with spaces, or use the asterisk (*) to indicate all users who are logged in to the local Routing Engine.

For the list of logging facilities and severity levels, see *Specifying the Facility and Severity of Messages to Include in the Log*. For information about the **match** statement, see [“Using Regular Expressions to Refine the Set of Logged Messages” on page 156](#).

- Related Documentation**
- *Single-Chassis System Logging Configuration Overview*
 - *Examples: Configuring System Logging*

Directing System Log Messages to the Console

To direct system log messages to the console of the local Routing Engine, include the **console** statement at the **[edit system syslog]** hierarchy level:

```

[edit system syslog]
console {
    facility severity;
}

```

For the list of logging facilities and severity levels, see *Specifying the Facility and Severity of Messages to Include in the Log*.

- Related Documentation**
- *Single-Chassis System Logging Configuration Overview*
 - *Examples: Configuring System Logging*

Disabling the System Logging of a Facility

To disable the logging of messages that belong to a particular facility, include the **facility none** statement in the configuration. This statement is useful when, for example, you want to log messages that have the same severity level and belong to all but a few facilities. Instead of including a statement for each facility you want to log, you can include the **any severity** statement and then a **facility none** statement for each facility that you do not want to log. For example, the following logs all messages at the **error** level or higher to the console, except for messages from the **daemon** and **kernel** facilities. Messages from those facilities are logged to the file **>/var/log/internals** instead:

```

[edit system syslog]
console {
    any error;
    daemon none;
    kernel none;
}
file internals {
    daemon info;
    kernel info;
}

```

}

Related Documentation

- *Single-Chassis System Logging Configuration Overview*

Displaying a Log File from a Single-Chassis System

To display a log file stored on a single-chassis system, enter Junos OS CLI operational mode and issue the following commands:

```
user@switch> show log log-filename
user@switch> file show log-file-pathname
```

By default, the commands display the file stored on the local Routing Engine.

The following example shows the output from the **show log messages** command:

```
user@switch1> show log messages
Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 1
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 2
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 3
...
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon memory usage (Management
process): new instance detected (variable: sysAppLElmtRunMemory.5.6.2293)
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon memory usage (Command-line
interface): new instance detected (variable: sysAppLElmtRunMemory.5.8.2292)
...
Nov  4 12:08:30 switch1 rpdf[957]: task_connect: task BGP_100.10.10.1.6+179 addr
10.10.1.6+179: Can't assign requested
address
Nov  4 12:08:30 switch1 rpdf[957]: bgp_connect_start: connect 10.10.1.6 (Internal
AS 100): Can't assign requested address
Nov  4 12:10:24 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'exit '
Nov  4 12:10:27 switch1 mgd[2293]: UI_DBASE_LOGOUT_EVENT: User 'jsmith' exiting
configuration mode
Nov  4 12:10:31 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'show log messages'
```

The following example shows the output from the **file show** command. The file in the pathname **/var/log/processes** has been previously configured to include messages from the daemon facility.

```
user@switch1> file show /var/log/processes
Feb 22 08:58:24 switch1 snmpd[359]: SNMPD_TRAP_WARM_START: trap_generate_warm:
SNMP trap: warm start
Feb 22 20:35:07 switch1 snmpd[359]: SNMPD_THROTTLE_QUEUE_DRAINED:
trap_throttle_timer_handler: cleared all throttled traps
Feb 23 07:34:56 switch1 snmpd[359]: SNMPD_TRAP_WARM_START: trap_generate_warm:
SNMP trap: warm start
Feb 23 07:38:19 switch1 snmpd[359]: SNMPD_TRAP_COLD_START: trap_generate_cold:
SNMP trap: cold start
...
```

- Related Documentation**
- [Interpreting Messages Generated in Standard Format on page 149](#)
 - [Interpreting Messages Generated in Structured-Data Format on page 146](#)

Including Priority Information in System Log Messages

The facility and severity level of a message are together referred to as its *priority*. By default, messages logged in the standard Junos OS format do not include information about priority. To include priority information in standard-format messages directed to a file, include the **explicit-priority** statement at the **[edit system syslog file *filename*]** hierarchy level:

```
[edit system syslog file filename]
facility severity;
explicit-priority;
```



NOTE: Messages logged in structured-data format include priority information by default. If you include the **structured-data** statement at the **[edit system syslog file *filename*]** hierarchy level along with the **explicit-priority** statement, the **explicit-priority** statement is ignored and messages are logged in structured-data format.

For information about the **structured-data** statement, see *Logging Messages in Structured-Data Format*. For information about the contents of a structured-data message, see the *Junos OS System Log Reference for Security Devices*.

To include priority information in messages directed to a remote machine or the other Routing Engine, include the **explicit-priority** statement at the **[edit system syslog host (*hostname* | other-routing-engine)]** hierarchy level:

```
[edit system syslog host (hostname | other-routing-engine)]
facility severity;
explicit-priority;
```



NOTE: The **other-routing-engine** option does not apply to the QFX Series.

The priority recorded in a message always indicates the original, local facility name. If the **facility-override** statement is included for messages directed to a remote destination, the Junos OS system logging utility still uses the alternative facility name for the messages themselves when directing them to the remote destination. For more information, see [“Changing the Alternative Facility Name for System Log Messages Directed to a Remote Destination” on page 155](#).

When the **explicit-priority** statement is included, the Junos OS logging utility prepends codes for the facility name and severity level to the message tag name, if the message has one:

FACILITY-severity[-TAG]

(The tag is a unique identifier assigned to some Junos OS system log messages.)

In the following example, the **CHASSISD_PARSE_COMPLETE** message belongs to the **daemon** facility and is assigned severity **info** (6):

```
Aug 21 12:36:30 router1 chassisd[522]: %DAEMON-6-CHASSISD_PARSE_COMPLETE:
Using new configuration
```

When the **explicit-priority** statement is not included, the priority does not appear in the message:

```
Aug 21 12:36:30 router1 chassisd[522]: CHASSISD_PARSE_COMPLETE: Using new
configuration
```

For more information about message formatting, see the *Junos OS System Log Reference for Security Devices*.

- Related Documentation**
- *Single-Chassis System Logging Configuration Overview*
 - *Examples: Configuring System Logging*

Including the Year or Millisecond in Timestamps

By default, the timestamp recorded in a standard-format system log message specifies the month, date, hour, minute, and second when the message was logged, as in the following example:

```
Aug 21 15:36:30
```

To include the year, the millisecond, or both, in the timestamp, include the **time-format** statement at the **[edit system syslog]** hierarchy level:

```
[edit system syslog]
time-format (year | millisecond | year millisecond);
```

The modified timestamp is used in messages directed to each destination configured by a **file**, **console**, or **user** statement at the **[edit system syslog]** hierarchy level, but not to destinations configured by a **host** statement.

The following example illustrates the format for a timestamp that includes both the millisecond (401) and the year (2010):

```
Aug 21 15:36:30.401 2010
```




NOTE: By default, messages logged in structured-data format include the year and millisecond. If you include the `structured-data` statement at the `[edit system syslog file filename]` hierarchy level along with the `time-format` statement, the `time-format` statement is ignored and messages are logged in structured-data format.

For information about the `structured-data` statement, see [“Logging Messages in Structured-Data Format” on page 145](#). For information about interpreting messages in a structured-data format, see [“Interpreting Messages Generated in Structured-Data Format” on page 146](#).

Logging Messages in Structured-Data Format

You can log messages to a file in structured-data format instead of the standard Junos OS format. The structured-data format provides more information without adding significant length, and makes it easier for automated applications to extract information from a message.

The structured-data format complies with Internet draft **draft-ietf-syslog-protocol-21.txt**. The draft establishes a standard message format regardless of the source or transport protocol for logged messages.

To output messages to a file in structured-data format, include the **structured-data** statement at the `[edit system syslog file filename]` hierarchy level:

```
[edit system syslog file filename]
  facility severity;
  structured-data {
    brief;
  }
```

The optional **brief** statement suppresses the English-language text that appears by default at the end of a message to describe the error or event. For information about the fields in a structured-data-format message, see [“Interpreting Messages Generated in Structured-Data Format” on page 146](#).

The structured format is used for all messages logged to the file that are generated by a Junos OS process or software library.



NOTE: If you include either or both of the `explicit-priority` and `time-format` statements along with the `structured-data` statement, they are ignored. These statements apply to the standard Junos OS system log format, not to structured-data format.

Interpreting Messages Generated in Structured-Data Format

By default, Junos OS processes and software libraries write messages to the system log file in structured-data format. For information about the **structured-data** statement, see *Logging Messages in Structured-Data Format*.

Structured-format makes it easier for automated applications to extract information from the message. In particular, the standardized format for reporting the value of variables (elements in the English-language message that vary depending on the circumstances that triggered the message) makes it easy for an application to extract those values.

The structured-data format for a message includes the following fields (which appear here on two lines only for legibility):

```
<priority code>version timestamp hostname process processID TAG [junos@2636.platform
variable-value-pairs] message-text
```

Table 30 on page 146 describes the fields. If the system logging utility cannot determine the value in a particular field, a hyphen (-) appears instead.

Table 30: Fields in Structured-Data Messages

| Field | Description | Examples |
|------------------------------|---|--|
| <priority code> | Number that indicates the facility and severity of a message. It is calculated by multiplying the facility number by 8 and then adding the numerical value of the severity. For a mapping of the numerical codes to facility and severity, see <i>Specifying the Facility and Severity of Messages to Include in the Log</i> . | <165> for a message from the pfe facility (facility=20) with severity notice (severity=5). |
| version | Version of the Internet Engineering Task Force (IETF) system logging protocol specification. | 1 for the initial version |
| timestamp | Time when the message was generated, in one of two representations: <ul style="list-style-type: none"> YYYY-MM-DDTHH:MM:SS.MSZ is the year, month, day, hour, minute, second and millisecond in Universal Coordinated Time (UTC) YYYY-MM-DDTHH:MM:SS.MS+/-HH:MM is the year, month, day, hour, minute, second and millisecond in local time; the hour and minute that follows the plus sign (+) or minus sign (-) is the offset of the local time zone from UTC | 2007-02-15T09:17:15.719Z is 9:17 AM UTC on 15 February 2007. 2007-02-15T01:17:15.719-08:00 is the same timestamp expressed as Pacific Standard Time in the United States. |
| hostname | Name of the host that originally generated the message. | switch1 |

Table 30: Fields in Structured-Data Messages (*continued*)

| Field | Description | Examples |
|-----------------------------|--|---|
| <i>process</i> | Name of the Junos OS process that generated the message. | mgd |
| <i>processID</i> | UNIX process ID (PID) of the Junos process that generated the message. | 3046 |
| <i>TAG</i> | Junos OS system log message tag, which uniquely identifies the message. | UI_DBASE_LOGOUT_EVENT |
| <i>junos@2636.platform</i> | An identifier for the type of hardware platform that generated the message. The junos@2636 prefix indicates that the platform runs the Junos OS. It is followed by a dot-separated numerical identifier for the platform type. | junos@2636.1.1.1.2.18 |
| <i>variable-value-pairs</i> | A variable-value pair for each element in the <i>message-text</i> string that varies depending on the circumstances that triggered the message. Each pair appears in the format <i>variable</i> = " <i>value</i> ". | username="regress" |
| <i>message-text</i> | English-language description of the event or error (omitted if the brief statement is included at the [edit system syslog file <i>filename</i> structured-data] hierarchy level). | User 'regress' exiting configuration mode |

By default, the structured-data version of a message includes English text at the end, as in the following example (which appears on multiple lines only for legibility):

```
<165>1 2007-02-15T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT
[junos@2636.1.1.1.2.18 username="regress"] User 'regress' exiting configuration mode
```

When the brief statement is included at the [edit system syslog file *filename* structured-data] hierarchy level, the English text is omitted, as in this example:

```
<165>1 2007-02-15T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT
[junos@2636.1.1.1.2.18 username="regress"]
```

Table 31 on page 148 maps the codes that appear in the *priority-code* field to facility and severity level.



NOTE: Not all of the facilities and severities listed in Table 31 on page 148 can be included in statements at the [edit system syslog] hierarchy level (some are used by internal processes). For a list of the facilities and severity levels that can be included in the configuration, see *Specifying the Facility and Severity of Messages to Include in the Log*.

Table 31: Facility and Severity Codes in the priority-code Field

| Facility (number) | Severity emergency | alert | critical | error | warning | notice | info | debug |
|-------------------------------|-----------------------|-------|----------|-------|---------|--------|------|-------|
| kernel (0) | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| user (1) | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| mail (2) | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| daemon (3) | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| authorization (4) | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| syslog (5) | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| printer (6) | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| news (7) | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| uucp (8) | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| clock (9) | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| authorization-private (10) | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| ftp (11) | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| ntp (12) | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 |
| security (13) | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| console (14) | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| local0 (16) | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| dfc (17) | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| local2 (18) | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 |
| firewall (19) | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| pfe (20) | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 |
| conflict-log (21) | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| change-log (22) | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 |
| interactive-commands (23) | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |

Interpreting Messages Generated in Standard Format

The syntax of a standard-format message generated by a Junos OS process or subroutine library depends on whether it includes priority information:

- When the **explicit-priority** statement is included at the `[edit system syslog file filename]` or `[edit system syslog host hostname]` hierarchy level, a system log message has the following syntax:

```
timestamp message-source: %facility-severity-TAG: message-text
```

- When directed to the console or to users, or when the **explicit-priority** statement is not included for files or remote hosts, a system log message has the following syntax:

```
timestamp message-source: TAG: message-text
```

Table 32 on page 149 describes the message fields.

Table 32: Fields in Standard-Format Messages

| Field | Description |
|-----------------------|---|
| <i>timestamp</i> | Time at which the message was logged. |
| <i>message-source</i> | Identifier of the process or component that generated the message and the routing platform on which the message was logged. This field includes two or more subfields: hostname, process and process ID (PID). If the process does not report its PID, the PID is not displayed. The message source subfields are displayed in the following format: <i>hostname process[process-ID]</i> |
| <i>facility</i> | Code that specifies the facility to which the system log message belongs. For a mapping of codes to facility names, see Table: Facility Codes Reported in Priority Information in “Including Priority Information in System Log Messages” on page 143. |
| <i>severity</i> | Numerical code that represents the severity level assigned to the system log message. For a mapping of codes to severity names, see Table: Numerical Codes for Severity Levels Reported in Priority Information in “Including Priority Information in System Log Messages” on page 143. |
| <i>TAG</i> | Text string that uniquely identifies the message, in all uppercase letters and using the underscore (_) to separate words. The tag name begins with a prefix that indicates the generating software process or library. The entries in this reference are ordered alphabetically by this prefix. Not all processes on a routing platform use tags, so this field does not always appear. |
| <i>message-text</i> | Text of the message. |

Specifying Log File Size, Number, and Archiving Properties

To prevent log files from growing too large, by default the Junos OS system logging utility writes messages to a sequence of files of a defined size. The files in the sequence are referred to as *archive* files to distinguish them from the *active* file to which messages are currently being written. The default maximum size depends on the platform type:

- 128 kilobytes (KB) for EX Series switches
- 1 megabyte (MB) for M Series, MX Series, and T Series routers
- 10 MB for TX Matrix or TX Matrix Plus routers
- 1 MB for the QFX Series

When an active log file called **logfile** reaches the maximum size, the logging utility closes the file, compresses it, and names the compressed archive file **logfile.0.gz**. The logging utility then opens and writes to a new active file called **logfile**. This process is also known as file rotation. When the new **logfile** reaches the configured maximum size, **logfile.0.gz** is renamed **logfile.1.gz**, and the new **logfile** is closed, compressed, and renamed **logfile.0.gz**. By default, the logging utility creates up to 10 archive files in this manner. When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the last archived file are overwritten by the current active file. The logging utility by default also limits the users who can read log files to the **root** user and users who have Junos OS **maintenance** permission.

Junos OS provides a configuration statement **log-rotate-frequency** that configures the system log file rotation frequency by configuring the time interval for checking the log file size. The frequency can be set to a value of 1 minute through 59 minutes. The default frequency is 15 minutes.

To configure the log rotation frequency, include the **log-rotate-frequency** statement at the **[edit system syslog]** hierarchy level.

You can include the **archive** statement to change the maximum size of each file, how many archive files are created, and who can read log files.

To configure values that apply to all log files, include the **archive** statement at the **[edit system syslog]** hierarchy level:

```
archive <files number> <size size> <world-readable | no-world-readable>;
```

To configure values that apply to a specific log file, include the **archive** statement at the **[edit system syslog file *filename*]** hierarchy level:

```
archive <archive-sites (ftp-url <password password>)> <files number> <size size>  
      <start-time "YYYY-MM-DD.hh:mm"> <transfer-interval minutes> <world-readable |  
      no-world-readable>;
```

archive-sites *site-name* specifies a list of archive sites that you want to use for storing files. The ***site-name*** value is any valid FTP URL to a destination. If more than one site name is configured, a list of archive sites for the system log files is created. When a file is archived, the router or switch attempts to transfer the file to the first URL in the list,

moving to the next site only if the transfer does not succeed. The log file is stored at the archive site with the specified log filename. For information about how to specify valid FTP URLs, see [Format for Specifying Filenames and URLs in Junos OS CLI Commands](#).

binary-data Mark file as containing binary data. This allows proper archiving of binary files, such as WTMP files (login records for UNIX based systems). To restore the default setting, include the **no-binary-data** statement.

files *number* specifies the number of files to create before the oldest file is overwritten. The value can be from 1 through 1000.

size *size* specifies the maximum size of each file. The value can be from 64 KB (64k) through 1 gigabyte (1g); to represent megabytes, use the letter **m** after the integer. There is no space between the digits and the **k**, **m**, or **g** units letter.

start-time "YYYY-MM-DD.hh:mm" defines the date and time in the local time zone for a one-time transfer of the active log file to the first reachable site in the list of sites specified by the **archive-sites** statement.

transfer-interval *interval* defines the amount of time the current log file remains open (even if it has not reached the maximum possible size) and receives new statistics before it is closed and transferred to an archive site. This interval value can be from 5 through 2880 minutes.

world-readable enables all users to read log files. To restore the default permissions, include the **no-world-readable** statement.

Related Documentation

- [Single-Chassis System Logging Configuration Overview](#)
- [Examples: Configuring System Logging](#)
- [Routing Matrix with a TX Matrix Plus Router Solutions Page](#)

Specifying the Facility and Severity of Messages to Include in the Log

Each system log message belongs to a *facility*, which is a group of messages that are either generated by the same software process or concern a similar condition or activity (such as authentication attempts). Each message is also preassigned a *severity level*, which indicates how seriously the triggering event affects router functions.

When you configure logging for a facility and destination, you specify a severity level for each facility. Messages from the facility that are rated at that level or higher are logged to the destination:

```
[edit system syslog]
(console | file filename | host destination | user username) {
    facility severity;
}
```

Related Documentation

- [Junos OS System Logging Facilities and Message Severity Levels on page 152](#)
- [Single-Chassis System Logging Configuration Overview](#)

- *Examples: Configuring System Logging*

Junos OS System Logging Facilities and Message Severity Levels

Table 33 on page 152 lists the Junos OS system logging facilities that you can specify in configuration statements at the **[edit system syslog]** hierarchy level.

Table 33: Junos OS System Logging Facilities

| Facility | Type of Event or Error |
|-----------------------------|---|
| any | All (messages from all facilities) |
| authorization | Authentication and authorization attempts |
| change-log | Changes to the Junos OS configuration |
| conflict-log | Specified configuration is invalid on the router type |
| daemon | Actions performed or errors encountered by system processes |
| dfc | Events related to dynamic flow capture |
| firewall | Packet filtering actions performed by a firewall filter |
| ftp | Actions performed or errors encountered by the FTP process |
| interactive-commands | Commands issued at the Junos OS command-line interface (CLI) prompt or by a client application such as a Junos XML protocol or NETCONF XML client |
| kernel | Actions performed or errors encountered by the Junos OS kernel |
| pfe | Actions performed or errors encountered by the Packet Forwarding Engine |
| user | Actions performed or errors encountered by user-space processes |

Table 34 on page 152 lists the severity levels that you can specify in configuration statements at the **[edit system syslog]** hierarchy level. The levels from **emergency** through **info** are in order from highest severity (greatest effect on functioning) to lowest.

Unlike the other severity levels, the **none** level disables logging of a facility instead of indicating how seriously a triggering event affects routing functions. For more information, see “Disabling the System Logging of a Facility” on page 141.

Table 34: System Log Message Severity Levels

| Severity Level | Description |
|----------------|------------------------------|
| any | Includes all severity levels |

Table 34: System Log Message Severity Levels (*continued*)

| Severity Level | Description |
|------------------|---|
| none | Disables logging of the associated facility to a destination |
| emergency | System panic or other condition that causes the router to stop functioning |
| alert | Conditions that require immediate correction, such as a corrupted system database |
| critical | Critical conditions, such as hard errors |
| error | Error conditions that generally have less serious consequences than errors at the emergency, alert, and critical levels |
| warning | Conditions that warrant monitoring |
| notice | Conditions that are not errors but might warrant special handling |
| info | Events or nonerror conditions of interest |

**Related
Documentation**

- *Single-Chassis System Logging Configuration Overview*
- *Examples: Configuring System Logging*

System Log Default Facilities for Messages Directed to a Remote Destination

Table 35 on page 153 lists the default alternative facility name next to the Junos OS-specific facility name for which it is used. For facilities that are not listed, the default alternative name is the same as the local facility name.

Table 35: Default Facilities for Messages Directed to a Remote Destination

| Junos OS-specific Local Facility | Default Facility When Directed to Remote Destination |
|----------------------------------|--|
| change-log | local6 |
| conflict-log | local5 |
| dfc | local1 |
| firewall | local3 |
| interactive-commands | local7 |
| pfe | local4 |

- Related Documentation**
- *Single-Chassis System Logging Configuration Overview*

Alternate Facilities for System Log Messages Directed to a Remote Destination

Table 36 on page 154 lists the facilities that you can specify in the **facility-override** statement.

Table 36: Facilities for the facility-override Statement

| Facility | Description |
|----------------------|---|
| authorization | Authentication and authorization attempts |
| daemon | Actions performed or errors encountered by system processes |
| ftp | Actions performed or errors encountered by the FTP process |
| kernel | Actions performed or errors encountered by the Junos OS kernel |
| local0 | Local facility number 0 |
| local1 | Local facility number 1 |
| local2 | Local facility number 2 |
| local3 | Local facility number 3 |
| local4 | Local facility number 4 |
| local5 | Local facility number 5 |
| local6 | Local facility number 6 |
| local7 | Local facility number 7 |
| user | Actions performed or errors encountered by user-space processes |

We do not recommend including the **facility-override** statement at the **[edit system syslog host other-routing-engine]** hierarchy level. It is not necessary to use alternative facility names when directing messages to the other Routing Engine, because its Junos OS system logging utility can interpret the Junos OS-specific names.

- Related Documentation**
- *Examples: Assigning an Alternative Facility to System Log Messages Directed to a Remote Destination*
 - *Single-Chassis System Logging Configuration Overview*

Changing the Alternative Facility Name for System Log Messages Directed to a Remote Destination

Some facilities assigned to messages logged on the local router or switch have Junos OS-specific names (see [Table 33 on page 152](#)). In the recommended configuration, a remote machine designated at the `[edit system syslog host hostname]` hierarchy level is not a Juniper Networks router or switch, so its syslogd utility cannot interpret the Junos OS-specific names. To enable the standard syslogd utility to handle messages from these facilities when messages are directed to a remote machine, a standard **localX** facility name is used instead of the Junos OS-specific facility name.

[Table 35 on page 153](#) lists the default alternative facility name next to the Junos OS-specific facility name it is used for.

The syslogd utility on a remote machine handles all messages that belong to a facility in the same way, regardless of the source of the message (the Juniper Networks router or switch or the remote machine itself). For example, the following statements in the configuration of the router called **local-router** direct messages from the **authorization** facility to the remote machine *monitor.mycompany.com*:

```
[edit system syslog]
host monitor.mycompany.com {
  authorization info;
}
```

The default alternative facility for the local **authorization** facility is also **authorization**. If the syslogd utility on **monitor** is configured to write messages belonging to the **authorization** facility to the file `/var/log/auth-attempts`, then the file contains the messages generated when users log in to **local-router** and the messages generated when users log in to **monitor**. Although the name of the source machine appears in each system log message, the mixing of messages from multiple machines can make it more difficult to analyze the contents of the **auth-attempts** file.

To make it easier to separate the messages from each source, you can assign an alternative facility to all messages generated on **local-router** when they are directed to **monitor**. You can then configure the syslogd utility on **monitor** to write messages with the alternative facility to a different file from messages generated on **monitor** itself.

To change the facility used for all messages directed to a remote machine, include the **facility-override** statement at the `[edit system syslog host hostname]` hierarchy level:

```
[edit system syslog host hostname]
facility severity;
facility-override facility;
```

In general, it makes sense to specify an alternative facility that is not already in use on the remote machine, such as one of the **localX** facilities. On the remote machine, you must also configure the syslogd utility to handle the messages in the desired manner.

[Table 36 on page 154](#) lists the facilities that you can specify in the **facility-override** statement.

We do not recommend including the **facility-override** statement at the **[edit system syslog host other-routing-engine]** hierarchy level. It is not necessary to use alternative facility names when directing messages to the other Routing Engine, because its Junos OS system logging utility can interpret the Junos OS-specific names.

The following example shows how to log all messages generated on the local router at the error level or higher to the local0 facility on the remote machine called monitor.mycompany.com:

```
[edit system syslog]
host monitor.mycompany.com {
  any error;
  facility-override local0;
}
```

The following example shows how to configure routers located in California and routers located in New York to send messages to a single remote machine called central-logger.mycompany.com. The messages from California are assigned to alternative facility local0 and the messages from New York are assigned to alternative facility local2.

- Configure California routers to aggregate messages in the local0 facility:

```
[edit system syslog]
host central-logger.mycompany.com {
  change-log info;
  facility-override local0;
}
```

- Configure New York routers to aggregate messages in the local2 facility:

```
[edit system syslog]
host central-logger.mycompany.com {
  change-log info;
  facility-override local2;
}
```

On central-logger, you can then configure the system logging utility to write messages from the local0 facility to the file **change-log** and the messages from the local2 facility to the file **new-york-config**.

Related Documentation

- [Table 35 on page 153](#)
- [Alternate Facilities for System Log Messages Directed to a Remote Destination on page 154](#)
- *Examples: Assigning an Alternative Facility to System Log Messages Directed to a Remote Destination*

Using Regular Expressions to Refine the Set of Logged Messages

The predefined facilities group together related messages, but you can also use regular expression matching to specify more exactly which messages from a facility are logged to a file, a user terminal, or a remote destination.

To specify the text string that must (or must not) appear in a message for the message to be logged to a destination, include the **match** statement and specify the regular expression which the text string must match:

```
match "regular-expression";
```

You can include this statement at the following hierarchy levels:

- **[edit system syslog file *filename*]** (for a file)
- **[edit system syslog user (*username* | *)]** (for a specific user session or for all user sessions on a terminal)
- **[edit system syslog host (*hostname* | other-routing-engine)]** (for a remote destination)

In specifying the regular expression, use the notation defined in POSIX Standard 1003.2 for extended (modern) UNIX regular expressions. Explaining regular expression syntax is beyond the scope of this document, but POSIX standards are available from the Institute of Electrical and Electronics Engineers (IEEE, <http://www.ieee.org>).

Table 37 on page 157 specifies which character or characters are matched by some of the regular expression operators that you can use in the match statement. In the descriptions, the term *term* refers to either a single alphanumeric character or a set of characters enclosed in square brackets, parentheses, or braces.



NOTE: The match statement is not case-sensitive.

Table 37: Regular Expression Operators for the match Statement

| Operator | Matches |
|-----------------------|---|
| . (period) | One instance of any character except the space. |
| * (asterisk) | Zero or more instances of the immediately preceding term. |
| + (plus sign) | One or more instances of the immediately preceding term. |
| ? (question mark) | Zero or one instance of the immediately preceding term. |
| (pipe) | One of the terms that appears on either side of the pipe operator. |
| ! (exclamation point) | Any string except the one specified by the expression, when the exclamation point appears at the start of the expression. Use of the exclamation point is Junos OS-specific. |
| ^ (caret) | Start of a line, when the caret appears outside square brackets. One instance of any character that does not follow it within square brackets, when the caret is the first character inside square brackets. |
| \$ (dollar sign) | End of a line. |

Table 37: Regular Expression Operators for the match Statement (*continued*)

| Operator | Matches |
|------------------------------|--|
| [] (paired square brackets) | One instance of one of the enclosed alphanumeric characters. To indicate a range of characters, use a hyphen (-) to separate the beginning and ending characters of the range. For example, [a-z0-9] matches any letter or number. |
| () (paired parentheses) | One instance of the evaluated value of the enclosed term. Parentheses are used to indicate the order of evaluation in the regular expression. |

Using Regular Expressions

Filter messages that belong to the **interactive-commands** facility, directing those that include the string **configure** to the terminal of the root user:

```
[edit system syslog]
user root {
  interactive-commands any;
  match ".*configure.*";
}
```

Messages like the following appear on the **root** user's terminal when a user issues a **configure** command to enter configuration mode:

```
timestamp router-name mgd[PID]: UI_CMDLINE_READ_LINE: User 'user', command
'configure private'
```

Filter messages that belong to the **daemon** facility and have a severity of **error** or higher, directing them to the file **/var/log/process-errors**. Omit messages generated by the SNMP process (snmpd), instead directing them to the file **/var/log/snmpd-errors**:

```
[edit system syslog]
file process-errors {
  daemon error;
  match "!(.*snmpd.*)";
}
file snmpd-errors {
  daemon error;
  match ".*snmpd.*";
}
```

Related Documentation

- *Single-Chassis System Logging Configuration Overview*
- *Examples: Configuring System Logging*

CHAPTER 8

Configuration Tasks for Network Management

- [Configuring Console and Auxiliary Port Properties on page 159](#)
- [Configuring SSH Service for Remote Access to the Router or Switch on page 160](#)
- [Configuring Telnet Service for Remote Access to a Switch on page 162](#)

Configuring Console and Auxiliary Port Properties

The console port and auxiliary port on a switch provide out-of-band remote access to the switch. You can configure the console and auxiliary ports so that an external data terminal may be connected to the switch. The console port is enabled by default. The console port speed is 9600 baud, except on OCX Series devices, on which it is 115200 baud. The auxiliary port is disabled by default.

By default, terminal connections to the console and auxiliary ports are secure. When you configure the console and auxiliary ports as insecure, root logins are not allowed to establish terminal connections, and superusers and anyone with a user identifier (UID) of 0 are not allowed to establish terminal connections in multiuser mode.

To configure the console and auxiliary port properties on the switch:

1. To specify that the console port session should terminate if the connection to the data carrier is lost:

```
[edit system ports]
user@switch# set console log-out-on-disconnect
```

2. To specify the auxiliary port terminal type:

```
[edit system ports]
user@switch# set auxiliary type (ansi | small-xterm | vt100 | xterm)
```

For example, to specify the auxiliary port terminal type of **xterm** with a display of 80 columns by 65 rows:

```
[edit system ports]
user@switch# set auxiliary type xterm
```

3. To check the configuration:

```
[edit system ports]
```

```
user@switch# show
console log-out-on-disconnect;
auxiliary type xterm;
```

- Related Documentation**
- *auxiliary*
 - *console (Physical Port)*
 - *ports*

Configuring SSH Service for Remote Access to the Router or Switch

To configure the router or switch to accept SSH as an access service, include the **ssh** statement at the **[edit system services]** hierarchy level:

```
[edit system services]
ssh {
  ciphers [ cipher-1 cipher-2 cipher-3 ...]
  client-alive-count-max number;
  client-alive-interval seconds;
  connection-limit limit;
  hostkey-algorithm <algorithm | no-algorithm>;
  key-exchange algorithm;
  macs algorithm;
  max-sessions-per-connection number;
  no-passwords;
  no-tcp-forwarding;
  protocol-version [v1 v2] ;
  rate-limit limit;
  root-login <allow | deny | deny-password>;
}
```

By default, the router or switch supports a limited number of simultaneous SSH sessions and connection attempts per minute. Use the following statements to change the defaults:

- **connection-limit *limit***—Maximum number of simultaneous connections per protocol (IPv4 and IPv6). The range is a value from 1 through 250. The default is 75. When you configure a connection limit, the limit is applicable to the number of SSH sessions per protocol (IPv4 and IPv6). For example, a connection limit of 10 allows 10 IPv6 SSH sessions and 10 IPv4 SSH sessions.
- **max-sessions-per-connection *number***—Include this statement to specify the maximum number of SSH sessions allowed per single SSH connection. This allows you to limit the number of cloned sessions tunneled within a single SSH connection. The default value is 10.
- **rate-limit *limit***—Maximum number of connection attempts accepted per minute (a value from 1 through 250). The default is 150. When you configure a rate limit, the limit is applicable to the number of connection attempts per protocol (IPv4 and IPv6). For example, a rate limit of 10 allows 10 IPv6 SSH session connection attempts per minute and 10 IPv4 SSH session connection attempts per minute.

By default, a user can create an SSH tunnel over a CLI session to a router running Junos OS via SSH. This type of tunnel could be used to forward TCP traffic, bypassing any firewall filters or ACLs, allowing access to resources beyond the router. Use the **no-tcp-forwarding** option to prevent a user from creating an SSH tunnel to a router via SSH.

For information about other configuration settings, see the following topics:

- [Configuring the Root Login Through SSH on page 161](#)
- [Configuring the SSH Protocol Version on page 161](#)
- [Configuring the Client Alive Mechanism on page 161](#)

Configuring the Root Login Through SSH

By default, users are allowed to log in to the router or switch as **root** through SSH. To control user access through SSH, include the **root-login** statement at the **[edit system services ssh]** hierarchy level:

```
[edit system services ssh]
root-login (allow | deny | deny-password);
```

allow—Allows users to log in to the router or switch as root through SSH. The default is **allow**.

deny—Disables users from logging in to the router or switch as root through SSH.

deny-password—Allows users to log in to the router or switch as root through SSH when the authentication method (for example, RSA) does not require a password.

Configuring the SSH Protocol Version

By default, only version 2 of the SSH protocol is enabled. To enable version 1, you must explicitly configure it.

To configure the router or switch to use version 1 and 2 of the SSH protocol, include the **protocol-version** statement and specify **v1** and **v2** at the **[edit system services ssh]** hierarchy level:

```
[edit system services ssh]
protocol-version [ v1 v2 ];
```

To configure the router or switch to use only version 1 of the SSH protocol, include the **protocol-version** statement and specify **v1** at the **[edit system services ssh]** hierarchy level:

```
[edit system services ssh]
protocol-version [ v1 ];
```

For J Series Services Routers, the export license software supports SSH version 1 only.

Configuring the Client Alive Mechanism

The client alive mechanism is valuable when the client or server depends on knowing when a connection has become inactive. It differs from the standard keepalive mechanism because the client alive messages are sent through the encrypted channel. The client

alive mechanism is not enabled at default. To enable it, configure the **client-alive-count-max** and the **client-alive-interval**. This option applies to SSH protocol version 2 only.

In the following example, unresponsive SSH clients will be disconnected after approximately 100 seconds (20 x 5).

```
[edit system services ssh]
client-alive-count-max 5;
client-alive-interval 20;
```

Configuring Telnet Service for Remote Access to a Switch

Telnet provides unencrypted access to network devices. Configuring Telnet service for a switch enables in-band remote access to the switch.

By default, the switch supports a limited number of simultaneous Telnet sessions and connection attempts per minute. Optionally, you can change the default Telnet settings by configuring the connection limit and rate limit at the **[edit system services telnet]** hierarchy level.

The connection limit is the maximum number of simultaneous connections per protocol (IPv4). The range is from 1 through 250. The default is 75.

The rate limit is the maximum number of connection attempts accepted per minute per protocol. The range is from 1 through 250. The default is 150.

To configure Telnet service:

1. To specify the connection limit:

```
[edit system services]
user@switch# set telnet connection-limit connection-limit
```

2. To specify the rate limit:

```
[edit system services]
user@switch# set telnet rate-limit rate-limit
```

3. Check that the Telnet connection limit and rate limit show the values you specified:

```
[edit system services]
user@switch# show
telnet {
  connection-limit 50;
  rate-limit 100;
}
```

Related Documentation

- *Understanding Telnet on the QFabric System*
- *Example: Limiting the Number of Login Attempts for SSH and Telnet Sessions to Prevent Unauthorized Access*

PART 3

Monitoring and Troubleshooting

- [Monitoring on page 165](#)
- [Troubleshooting on page 181](#)

CHAPTER 9

Monitoring

- [Displaying a Log File from a Single-Chassis System on page 165](#)
- [Monitoring Traffic Through the Router or Switch on page 166](#)
- [Monitoring RMON MIB Tables on page 169](#)
- [Monitoring SNMP on page 169](#)
- [Monitoring System Log Messages on page 171](#)
- [Pinging Hosts on page 172](#)
- [Tracing SNMP Activity on a Device Running Junos OS on page 173](#)
- [Using the Enterprise-Specific Utility MIB to Enhance SNMP Coverage on page 176](#)
- [Displaying Commit Script Output on page 178](#)

Displaying a Log File from a Single-Chassis System

To display a log file stored on a single-chassis system, enter Junos OS CLI operational mode and issue the following commands:

```
user@switch> show log log-filename
user@switch> file show log-file-pathname
```

By default, the commands display the file stored on the local Routing Engine.

The following example shows the output from the **show log messages** command:

```
user@switch1> show log messages
Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 1
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 2
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 3
...
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon memory usage (Management
process): new instance detected (variable: sysApp1ElmtRunMemory.5.6.2293)
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon memory usage (Command-line
interface): new instance detected (variable: sysApp1ElmtRunMemory.5.8.2292)
...
Nov  4 12:08:30 switch1 rpdf[957]: task_connect: task BGP_100.10.10.1.6+179 addr
```

```

10.10.1.6+179: Can't assign requested
address
Nov  4 12:08:30 switch1 rpdf[957]: bgp_connect_start: connect 10.10.1.6 (Internal
AS 100): Can't assign requested address
Nov  4 12:10:24 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'exit '
Nov  4 12:10:27 switch1 mgd[2293]: UI_DBASE_LOGOUT_EVENT: User 'jsmith' exiting
configuration mode
Nov  4 12:10:31 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'show log messages

```

The following example shows the output from the **file show** command. The file in the pathname **/var/log/processes** has been previously configured to include messages from the daemon facility.

```

user@switch1> file show /var/log/processes
Feb 22 08:58:24 switch1 snmpd[359]: SNMPD_TRAP_WARM_START: trap_generate_warm:
SNMP trap: warm start
Feb 22 20:35:07 switch1 snmpd[359]: SNMPD_THROTTLE_QUEUE_DRAINED:
trap_throttle_timer_handler: cleared all throttled traps
Feb 23 07:34:56 switch1 snmpd[359]: SNMPD_TRAP_WARM_START: trap_generate_warm:
SNMP trap: warm start
Feb 23 07:38:19 switch1 snmpd[359]: SNMPD_TRAP_COLD_START: trap_generate_cold:
SNMP trap: cold start
...

```

Related Documentation

- [Interpreting Messages Generated in Standard Format on page 149](#)
- [Interpreting Messages Generated in Structured-Data Format on page 146](#)

Monitoring Traffic Through the Router or Switch

To help with the diagnosis of a problem, display real-time statistics about the traffic passing through physical interfaces on the router or switch.

To display real-time statistics about physical interfaces, perform these tasks:

1. [Displaying Real-Time Statistics About All Interfaces on the Router or Switch on page 166](#)
2. [Displaying Real-Time Statistics About an Interface on the Router or Switch on page 167](#)

Displaying Real-Time Statistics About All Interfaces on the Router or Switch

Purpose Display real-time statistics about traffic passing through all interfaces on the router or switch.

Action To display real-time statistics about traffic passing through all interfaces on the router or switch:

```
user@host> monitor interface traffic
```

Sample Output

```

user@host> monitor interface traffic
host name                Seconds:15                Time: 12:31:09
Interface    Link    Input packets    (pps)    Output packets    (pps)
so-1/0/0     Down           0          (0)           0          (0)

```

| | | | | | |
|----------|------|--------|-----|-------|-----|
| so-1/1/0 | Down | 0 | (0) | 0 | (0) |
| so-1/1/1 | Down | 0 | (0) | 0 | (0) |
| so-1/1/2 | Down | 0 | (0) | 0 | (0) |
| so-1/1/3 | Down | 0 | (0) | 0 | (0) |
| t3-1/2/0 | Down | 0 | (0) | 0 | (0) |
| t3-1/2/1 | Down | 0 | (0) | 0 | (0) |
| t3-1/2/2 | Down | 0 | (0) | 0 | (0) |
| t3-1/2/3 | Down | 0 | (0) | 0 | (0) |
| so-2/0/0 | Up | 211035 | (1) | 36778 | (0) |
| so-2/0/1 | Up | 192753 | (1) | 36782 | (0) |
| so-2/0/2 | Up | 211020 | (1) | 36779 | (0) |
| so-2/0/3 | Up | 211029 | (1) | 36776 | (0) |
| so-2/1/0 | Up | 189378 | (1) | 36349 | (0) |
| so-2/1/1 | Down | 0 | (0) | 18747 | (0) |
| so-2/1/2 | Down | 0 | (0) | 16078 | (0) |
| so-2/1/3 | Up | 0 | (0) | 80338 | (0) |
| at-2/3/0 | Up | 0 | (0) | 0 | (0) |
| at-2/3/1 | Down | 0 | (0) | 0 | (0) |

Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D

Meaning The sample output displays traffic data for active interfaces and the amount that each field has changed since the command started or since the counters were cleared by using the **C** key. In this example, the **monitor interface** command has been running for 15 seconds since the command was issued or since the counters last returned to zero.

Displaying Real-Time Statistics About an Interface on the Router or Switch

Purpose Display real-time statistics about traffic passing through an interface on the router or switch.

Action To display traffic passing through an interface on the router or switch, use the following Junos OS CLI operational mode command:

```
user@host> monitor interface interface-name
```

Sample Output

```
user@host> monitor interface so-0/0/1
Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'
R1
Interface: so-0/0/1, Enabled, Link is Up
Encapsulation: PPP, Keepalives, Speed: 0C3 Traffic statistics:
  Input bytes:          5856541 (88 bps)
  Output bytes:         6271468 (96 bps)
  Input packets:        157629 (0 pps)
  Output packets:       157024 (0 pps)
Encapsulation statistics:
  Input keepalives:      42353
  Output keepalives:     42320
  LCP state: Opened
Error statistics:
  Input errors:          0
  Input drops:           0
  Input framing errors:  0
  Input runs:            0
  Input giants:          0
  Policed discards:      0
  L3 incompletes:        0
```

```

L2 channel errors:                0
L2 mismatch timeouts:            0
Carrier transitions:              1
Output errors:                   0
Output drops:                    0
Aged packets:                    0
Active alarms : None
Active defects: None
SONET error counts/seconds:
  LOS count                      1
  LOF count                      1
  SEF count                      1
  ES-S                          77
  SES-S                          77
SONET statistics:
  BIP-B1                        0
  BIP-B2                        0
  REI-L                        0
  BIP-B3                        0
  REI-P                        0
Received SONET overhead:  F1      : 0x00  J0      : 0xZ

```

Meaning The sample output shows the input and output packets for a particular SONET interface (**so-0/0/1**). The information can include common interface failures, such as SONET/SDH and T3 alarms, loopbacks detected, and increases in framing errors. For more information, see *Checklist for Tracking Error Conditions*.

To control the output of the command while it is running, use the keys shown in [Table 38 on page 168](#).

Table 38: Output Control Keys for the monitor interface Command

| Action | Key |
|--|----------|
| Display information about the next interface. The monitor interface command scrolls through the physical or logical interfaces in the same order that they are displayed by the show interfaces terse command. | N |
| Display information about a different interface. The command prompts you for the name of a specific interface. | I |
| Freeze the display, halting the display of updated statistics. | F |
| Thaw the display, resuming the display of updated statistics. | T |
| Clear (zero) the current delta counters since monitor interface was started. It does not clear the accumulative counter. | C |
| Stop the monitor interface command. | Q |

See the [CLI Explorer](#) for details on using match conditions with the **monitor traffic** command.

Monitoring RMON MIB Tables

Purpose Monitor remote monitoring (RMON) alarm, event, and log tables.

Action To display the RMON tables:

```
user@switch> show snmp rmon
Alarm
Index  Variable description                               Value State
      5  monitor
      jnxOperatingCPU.9.1.0.0                        5 falling threshold

Event
Index  Type                               Last Event
  1    log and trap                     2010-07-10 11:34:17 PDT
Event Index: 1
  Description: Event 1 triggered by Alarm 5, rising threshold (90) crossed,
(variable: jnxOperatingCPU.9.1.0.0, value: 100)
  Time: 2010-07-10 11:34:07 PDT
  Description: Event 1 triggered by Alarm 5, falling threshold (75) crossed,
(variable: jnxOperatingCPU.9.1.0.0, value: 5)
  Time: 2010-07-10 11:34:17 PDT
```

Meaning The display shows that an alarm has been defined to monitor jnxRmon MIB object jnxOperatingCPU, which represents the CPU utilization of the Routing Engine. The alarm is configured to generate an event that sends an SNMP trap and adds an entry to the logTable in the RMON MIB. The log table shows that two occurrences of the event have been generated—one for rising above a threshold of 90 percent, and one for falling below a threshold of 75 percent.

Related Documentation

- [Configuring RMON Alarms and Events on page 121](#)
- *show snmp rmon*
- *show snmp rmon history*
- *clear snmp statistics*
- *clear snmp history*

Monitoring SNMP

There are several commands that you can access in Junos OS operational mode to monitor SNMP information. Some of the commands are:

- **show snmp health-monitor**, which displays the health monitor log and alarm information.
- **show snmp mib**, which displays information from the MIBs, such as device and system information.

- **show snmp statistics**, which displays SNMP statistics such as the number of packets, silent drops, and invalid output values.
- **show snmp rmon**, which displays the RMON alarm, event, history, and log information

The following example provides sample output from the **show snmp health-monitor** command:

```
user@switch> show snmp health-monitor
Alarm
Index  Variable description                               Value State

32768 Health Monitor: root file system utilization
      jnxHrStoragePercentUsed.1                      58 active

32769 Health Monitor: /config file system utilization
      jnxHrStoragePercentUsed.2                      0 active

32770 Health Monitor: RE 0 CPU utilization
      jnxOperatingCPU.9.1.0.0                       0 active

32773 Health Monitor: RE 0 Memory utilization
      jnxOperatingBuffer.9.1.0.0                    35 active

32775 Health Monitor: jkernel daemon CPU utilization
      Init daemon                                   0 active
      Chassis daemon                               50 active
      Firewall daemon                              0 active
      Interface daemon                             5 active
      SNMP daemon                                  11 active
      MIB2 daemon                                  42 active
      ...
```

The following example provides sample output from the **show snmp mib** command:

```
user@switch> show snmp mib walk system

sysDescr.0    = Juniper Networks, Inc. qfx3500s internet router, kernel
JUNOS 11.1-20100926.0 #0: 2010-09-26 06:17:38 UTC builder@abc.juniper.net:
/volume/build/junos/11.1/production/20100926.0/obj-xlr/bsd/sys/compile/JUNIPER-xxxxx

Build date: 2010-09-26 06:00:10 U
sysObjectID.0 = jnxProductQFX3500
sysUpTime.0   = 24444184
sysContact.0  = J Smith
sysName.0     = Lab QFX3500
sysLocation.0 = Lab
sysServices.0 = 4
```

The following example provides sample output from the **show snmp statistics** command:

```
user@switch> show snmp statistics

SNMP statistics:
Input:
  Packets: 0, Bad versions: 0, Bad community names: 0,
  Bad community uses: 0, ASN parse errors: 0,
  Too big: 0, No such names: 0, Bad values: 0,
  Read only: 0, General errors: 0,
```

```

Total request varbinds: 0, Total set varbinds: 0,
Get requests: 0, Get nexts: 0, Set requests: 0,
Get responses: 0, Traps: 0,
Silent drops: 0, Proxy drops: 0, Commit pending drops: 0,
Throttle drops: 0, Duplicate request drops: 0
Output:
Packets: 0, Too bigs: 0, No such names: 0,
Bad values: 0, General errors: 0,
Get requests: 0, Get nexts: 0, Set requests: 0,
Get responses: 0, Traps: 0

```

- Related Documentation**
- *health-monitor*
 - *show snmp mib*
 - *show snmp statistics*

Monitoring System Log Messages

Purpose Display system log messages about the QFX Series. By looking through a system log file for any entries pertaining to the interface that you are interested in, you can further investigate a problem with an interface on the switch.

Action To view system log messages:

```
user@switch1> show log messages
```

Sample Output

```

Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:01 switch1 newsyslog[2283]: logfile turned over due to size>128K
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 1
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 2
Nov  4 11:30:06 switch1 chassism[952]: CM ENV Monitor: set fan speed is 65 percent
for Fan 3
...
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon
memory usage (Management process): new instance detected (variable:
sysApp1ElmtRunMemory.5.6.2293)
Nov  4 11:52:53 switch1 snmpd[944]: SNMPD_HEALTH_MON_INSTANCE: Health Monitor:
jroute daemon
memory usage (Command-line interface): new instance detected (variable:
sysApp1ElmtRunMemory.5.8.2292)
...
Nov  4 12:10:24 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'exit '
Nov  4 12:10:27 switch1 mgd[2293]: UI_DBASE_LOGOUT_EVENT: User 'jsmith' exiting
configuration mode
Nov  4 12:10:31 switch1 mgd[2293]: UI_CMDLINE_READ_LINE: User 'jsmith', command
'show log messages

```

Meaning The sample output shows the following entries in the **messages** file:

- A new log file was created when the previous file reached the maximum size of 128 kilobytes (KB).
- The fan speed for Fan 1, 2, and 3 is set at 65 percent.
- Health monitoring activity is detected.
- CLI commands were entered by the user jsmith.

Related Documentation

- [Overview of Junos OS System Log Messages on page 132](#)
- *Understanding the Implementation of System Log Messages on the QFabric System*
- *Example: Configuring System Log Messages*
- *clear log*
- *show log*
- *syslog*

Pinging Hosts

- Purpose** Use the CLI **ping** command to verify that a host can be reached over the network. This command is useful for diagnosing host and network connectivity problems. The switch sends a series of Internet Control Message Protocol (ICMP) echo (ping) requests to a specified host and receives ICMP echo responses.
- Action** To use the **ping** command to send four requests (ping count) to host3:
ping *host count number*

Sample Output

```
ping host3 count 4
user@switch> ping host3 count 4
PING host3.site.net (176.26.232.111): 56 data bytes
64 bytes from 176.26.232.111: icmp_seq=0 ttl=122 time=0.661 ms
64 bytes from 176.26.232.111: icmp_seq=1 ttl=122 time=0.619 ms
64 bytes from 176.26.232.111: icmp_seq=2 ttl=122 time=0.621 ms
64 bytes from 176.26.232.111: icmp_seq=3 ttl=122 time=0.634 ms

--- host3.site.net ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.619/0.634/0.661/0.017 ms
```

- Meaning**
- The **ping** results show the following information:
 - Size of the ping response packet (in bytes).
 - IP address of the host from which the response was sent.
 - Sequence number of the ping response packet. You can use this value to match the ping response to the corresponding ping request.
 - Time-to-live (ttl) hop-count value of the ping response packet.

- Total time between the sending of the ping request packet and the receiving of the ping response packet, in milliseconds. This value is also called round-trip time.
- Number of ping requests (probes) sent to the host.
- Number of ping responses received from the host.
- Packet loss percentage.
- Round-trip time statistics: minimum, average, maximum, and standard deviation of the round-trip time.

**Related
Documentation**

- *Troubleshooting Overview*
- *Understanding Troubleshooting Resources*

Tracing SNMP Activity on a Device Running Junos OS

SNMP tracing operations track activity for SNMP agents and record the information in log files. The logged error descriptions provide detailed information to help you solve problems faster.

By default, Junos OS does not trace any SNMP activity. If you include the **traceoptions** statement at the **[edit snmp]** hierarchy level, the default tracing behavior is:

- Important activities are logged in files located in the **/var/log** directory. Each log is named after the SNMP agent that generates it. Currently, the following log files are created in the **/var/log** directory when the **traceoptions** statement is used:
 - `chassisd`
 - `craftd`
 - `ilmid`
 - `mib2d`
 - `rmopd`
 - `serviced`
 - `snmpd`
- When a trace file named **filename** reaches its maximum size, it is renamed **filename.0**, then **filename.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. (For more information about how log files are created, see the *System Log Messages*.)
- Log files can be accessed only by the user who configured the tracing operation.

You cannot change the directory (**/var/log**) in which trace files are located. However, you can customize the other trace file settings by including the following statements at the **[edit snmp]** hierarchy level:

```
[edit snmp]
traceoptions {
```

```
file <files number> <match regular-expression> <size size> <world-readable |  
no-world-readable>;  
flag flag;  
memory-trace;  
no-remote-trace;  
no-default-memory-trace;  
}
```

These statements are described in the following sections:

- [Configuring the Number and Size of SNMP Log Files on page 174](#)
- [Configuring Access to the Log File on page 174](#)
- [Configuring a Regular Expression for Lines to Be Logged on page 175](#)
- [Configuring the Trace Operations on page 175](#)

Configuring the Number and Size of SNMP Log Files

By default, when the trace file reaches 128 kilobytes (KB) in size, it is renamed *filename.0*, then *filename.1*, and so on, until there are three trace files. Then the oldest trace file (*filename.2*) is overwritten.

You can configure the limits on the number and size of trace files by including the following statements at the **[edit snmp traceoptions]** hierarchy level:

```
[edit snmp traceoptions]  
file files number size size;
```

For example, set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation (*filename*) reaches 2 MB, *filename* is renamed *filename.0*, and a new file called *filename* is created. When the new *filename* reaches 2 MB, *filename.0* is renamed *filename.1* and *filename* is renamed *filename.0*. This process repeats until there are 20 trace files. Then the oldest file (*filename.19*) is overwritten by the newest file (*filename.0*).

The number of files can be from 2 through 1000 files. The file size of each file can be from 10 KB through 1 gigabyte (GB).

Configuring Access to the Log File

By default, log files can be accessed only by the user who configured the tracing operation.

To specify that any user can read all log files, include the **file world-readable** statement at the **[edit snmp traceoptions]** hierarchy level:

```
[edit snmp traceoptions]  
file world-readable;
```

To explicitly set the default behavior, include the **file no-world-readable** statement at the **[edit snmp traceoptions]** hierarchy level:

```
[edit snmp traceoptions]  
file no-world-readable;
```

Configuring a Regular Expression for Lines to Be Logged

By default, the trace operation output includes all lines relevant to the logged activities.

You can refine the output by including the **match** statement at the **[edit snmp traceoptions file *filename*]** hierarchy level and specifying a regular expression (regex) to be matched:

```
[edit snmp traceoptions]
file filename match regular-expression;
```

Configuring the Trace Operations

By default, only important activities are logged. You can specify which trace operations are to be logged by including the following **flag** statement (with one or more tracing flags) at the **[edit snmp traceoptions]** hierarchy level:

```
[edit snmp traceoptions]
flag {
  all;
  configuration;
  database;
  events;
  general;
  interface-stats;
  nonvolatile-sets;
  pdu;
  policy;
  protocol-timeouts;
  routing-socket;
  server;
  subagent;
  timer;
  varbind-error;
}
```

Table 39 on page 175 describes the meaning of the SNMP tracing flags.

Table 39: SNMP Tracing Flags

| Flag | Description | Default Setting |
|------------------------|---|-----------------|
| all | Log all operations. | Off |
| configuration | Log reading of the configuration at the [edit snmp] hierarchy level. | Off |
| database | Log events involving storage and retrieval in the events database. | Off |
| events | Log important events. | Off |
| general | Log general events. | Off |
| interface-stats | Log physical and logical interface statistics. | Off |

Table 39: SNMP Tracing Flags (*continued*)

| Flag | Description | Default Setting |
|--------------------------|--|-----------------|
| nonvolatile-set | Log nonvolatile SNMP set request handling. | Off |
| pdu | Log SNMP request and response packets. | Off |
| policy | Log policy processing. | Off |
| protocol-timeouts | Log SNMP response timeouts. | Off |
| routing-socket | Log routing socket calls. | Off |
| server | Log communication with processes that are generating events. | Off |
| subagent | Log subagent restarts. | Off |
| timer | Log internal timer events. | Off |
| varbind-error | Log variable binding errors. | Off |

To display the end of the log for an agent, issue the **show log agentd | last** operational mode command:

```
[edit]
user@host# run show log agentd | last
```

where **agent** is the name of an SNMP agent.

Related Documentation

- [Configuring SNMP on a Device Running Junos OS](#)
- [Configuration Statements at the \[edit snmp\] Hierarchy Level](#)
- [Example: Tracing SNMP Activity](#)
- [Configuring SNMP on page 113](#)

Using the Enterprise-Specific Utility MIB to Enhance SNMP Coverage

Even though the Junos OS has built-in performance metrics and monitoring options, you might need to have customized performance metrics. To make it easier for you to monitor such customized data through a standard monitoring system, the Junos OS provides you with an enterprise-specific Utility MIB that can store such data and thus extend SNMP support for managing and monitoring the data of your choice.

The enterprise-specific Utility MIB provides you with container objects of the following types: **32-bit counters**, **64-bit counters**, **signed integers**, **unsigned integers**, and **octet strings**. You can use these container MIB objects to store the data that are otherwise not supported for SNMP operations. You can populate data for these objects either by using

CLI commands or with the help of Op scripts and an RPC API that can invoke the CLI commands.

The following CLI commands enable you to set and clear Utility MIB object values:

- `request snmp utility-mib set instance name object-type <counter | counter 64 | integer | string | unsigned integer> object-value value`
- `request snmp utility-mib clear instance name object-type <counter | counter 64 | integer | string | unsigned integer>`

The `instance name` option of the `request snmp utility-mib <set | clear>` command specifies the name of the data instance and is the main identifier of the data. The `object-type <counter | counter 64 | integer | string | unsigned integer>` option enables you specify the object type, and the `object-value value` option enables you to set the value of the object.

To automate the process of populating Utility MIB data, you can use a combination of an event policy and event script. The following examples show the configuration for an event policy to run `show system buffers` every hour and to store the `show system buffers` data in Utility MIB objects by running an event script (`check-mbufs.slax`).

Event Policy Configuration

To configure an event policy that runs the `show system buffers` command every hour and invokes `check-mbufs.slax` to store the `show system buffers` data into Utility MIB objects, include the following statements at the `[edit]` hierarchy level:

```
event-options {
  generate-event {
    1-HOUR time-interval 3600;
  }
  policy Mbufs {
    events 1-HOUR;
    then {
      event-script check-mbufs.slax; # script stored at /var/db/scripts/event/
    }
  }
  event-script {
    file check-mbufs.slax;
  }
}
```

check-mbufs.slax Script

The following example shows the `check-mbufs.slax` script that is stored under `/var/db/scripts/event/`:

```
----- script START -----
version 1.0;

ns junos = "http://xml.juniper.net/junos/*/junos";
ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";
ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";
ns ext = "http://xmlsoft.org/XSLT/namespace";

match / {
  <op-script-results>{
    var $cmd = <command> "show system buffers";
    var $out = jcs:invoke($cmd);
```

```

var $lines = jcs:break_lines($out);
for-each ($lines) {
    if (contains(., "current/peak/max")) {
        var $pattern = "([0-9]+)/([0-9]+)/([0-9]+) mbufs";
        var $split = jcs:regex($pattern, .);
        var $result = $split[2];

        var $rpc = <request-snmp-utility-mib-set> {
            <object-type> "integer";
            <instance> "current-mbufs";
            <object-value> $result;
        }
        var $res = jcs:invoke($rpc);
    }
}
}
----- script END -----

```

You can run the following command to check the data stored in the Utility MIB as a result of the event policy and script shown in the preceding examples:

```

user@host> show snmp mib walk jnxUtilData ascii jnxUtilIntegerValue."current-mbufs"
= 0 jnxUtilIntegerTime."current-mbufs" = 07 da 05 0c 03 14 2c 00 2d 07 00
regress@caramels>

```



NOTE: The `show snmp mib walk` command is not available on the QFabric system, but you can use external SNMP client applications to perform this operation.

Related Documentation

- *Understanding SNMP Implementation in Junos OS*
- *Configuring SNMP on Devices Running Junos OS*
- *Monitoring SNMP Activity and Tracking Problems That Affect SNMP Performance on a Device Running Junos OS*
- *Optimizing the Network Management System Configuration for the Best Results*
- *Configuring Options on Managed Devices for Better SNMP Response Time*
- *Managing Traps and Informs*
- *Understanding the Implementation of SNMP on the QFabric System*

Displaying Commit Script Output

Table 40 on page 179 summarizes the Junos OS command-line interface (CLI) commands you can use to monitor and troubleshoot commit scripts. For more information about the `cscript.log` file, see *Tracing Commit Script Processing*.



NOTE: Tracing commit script processing, including the `cscript.log` file, is not supported on the QFX3000-G QFabric system.

Table 40: Commit Script Configuration and Operational Mode Commands

| Task | Command |
|---|--|
| Configuration Mode Commands | |
| Display errors and warnings generated by commit scripts. | commit or commit check |
| Display detailed information. | commit display detail |
| Display the underlying Extensible Markup Language (XML) data. | commit display xml |
| Display the postinheritance contents of the configuration database. This view includes transient changes, but does not include changes made in configuration groups. | show display commit-scripts |
| Display the postinheritance contents of the configuration database. This view excludes transient changes. | show display commit-scripts no-transients |
| Display the postinheritance configuration in XML format. Viewing the configuration in XML format can be helpful when you are writing XML Path Language (XPath) expressions and configuration element tags. | show display commit-scripts view |
| Display the postinheritance configuration in XML format, but exclude transient changes. | show display commit-scripts view display commit-scripts no-transients |
| Display all configuration groups data, including script-generated changes to the groups. | show groups display commit-scripts |
| Display a particular configuration group, including script-generated changes to the group. | show groups <i>group-name</i> display commit-scripts |
| Operational Mode Commands | |
| Display logging data associated with all commit script processing. | show log cscript.log |
| Display processing for only the most recent commit operation. | show log cscript.log last |
| Display processing for script errors. | show log cscript.log match error |

Table 40: Commit Script Configuration and Operational Mode Commands (*continued*)

| Task | Command |
|---|---|
| Display processing for a particular script. | show log cscript.log match <i>filename</i> |

Related Documentation

- *Tracing Commit Script Processing*

CHAPTER 10

Troubleshooting

- [Recovering from a Failed Software Installation on page 181](#)
- [Loading a Previous Configuration File on page 183](#)
- [Reverting to the Default Factory Configuration on page 184](#)
- [Reverting to the Rescue Configuration on page 184](#)
- [Recovering the Root Password on page 185](#)

Recovering from a Failed Software Installation

Problem **Description:** If the Junos OS appears to have been installed but the CLI does not work, or if the switch has no software installed, you can use this recovery installation procedure to install the Junos OS.

Solution If a Junos OS image already exists on the switch, you can either install the new Junos OS package in a separate partition, in which case both Junos OS images remain on the switch, or you can remove the existing Junos OS image before you start the new installation process.



NOTE: QFX5100, EX4600, and OCX Series switches do not have a separate partition to reinstall a Junos OS image.

A recovery image is created automatically on these switches. If a previously-running switch is powered on and unable to boot using a Junos OS image, you can boot the switch using the recovery Junos OS image by selecting an option in the “Select a recovery image” menu.

We suggest creating a system snapshot on your switch onto the external USB flash drive, and using the snapshot for recovery purposes. The system snapshot feature takes a “snapshot” of the files currently used to run the device—the complete contents of the /config directories, which include the running Juniper Networks Junos OS, the active configuration, and the rescue configuration, as well as the host OS—and copies all of these files into an external USB flash drive. See *Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Switch* or *Creating a Snapshot and Using It to Boot a Device*.

System snapshot is not supported on QFX10002 switches.

To perform a recovery installation:

1. Power on the switch. The loader script starts.
2. After the message **Loading /boot/defaults/loader.conf** appears, you are prompted with the following message:

Hit [Enter] to boot immediately, or space bar for command prompt.

Press the Spacebar to enter the manual loader. The **loader>** prompt appears.



NOTE: The loader prompt does not appear on QFX5100, QFX5200, EX4600, and OCX Series switches.

On QFX5100, QFX5200, EX4600, and OCX Series switches only, a recovery image is automatically saved if a previously-running switch is powered on and unable to boot using a Junos OS image.

The “Select a recovery image” menu appears on the console when one of these switches is booted and unable to load a version of Junos OS. Follow the instructions in the “Select a recovery image” menu to load the recovery version of Junos OS for one of these switches.

You can ignore the remainder of this procedure if you are using a QFX5100, QFX5200, EX4600, or OCX Series switch.

3. Enter the following command:

```
loader> install [- --format] [- --external] source
```

where:

- **format**—Enables you to erase the installation media before installing the installation package. If you do not include this option, the system installs the new Junos OS in a different partition from that of the most recently installed Junos OS.
- **external**—Installs the installation package onto external media (a USB stick, for example).
- **source**—Represents the name and location of the Junos OS package, either on a server on the network or as a file on an external media, as shown in the following two examples:
 - Network address of the server and the path on the server; for example, **tftp://192.171.28/junos/jinstall-qfx-5e-flex-15.1X53-D30.5-domestic-signed.tgz**
 - Junos OS package on a USB device (commonly stored in the root drive as the only file), for example, **file:///jinstall-qfx-5e-flex-15.1X53-D30.5-domestic-signed.tgz**.

The installation now proceeds normally and ends with a login prompt.

Related Documentation

- *Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Switch*
- *Creating a Snapshot and Using It to Boot a Device*

Loading a Previous Configuration File

You can use the **rollback <number>** command to return to a previously committed configuration file. A switch saves the last 50 committed configurations, including the rollback number, date, time, and name of the user who issued the **commit** configuration command.

Syntax

rollback <number>

Options

- **none**—Return to the most recently saved configuration.
- **number**—Return to the specified configuration.
 - **Range:** 0 through 49. The most recently saved configuration is number 0, and the oldest saved configuration is number 49.
 - **Default:** 0

To return to a configuration prior to the most recently committed one:

1. Specify the rollback number (here, 1 is entered and the configuration returns to the previously committed configuration 0):

[edit]

```
user@switch# rollback 1
load complete
```

2. Activate the configuration you have loaded:

```
[edit]
user@switch# commit
```

**Related
Documentation**

- [Configuration File Terms](#)

Reverting to the Default Factory Configuration

If for any reason the current active configuration fails, you can revert to the default factory configuration. The default factory configuration contains the basic configuration settings. This is the first configuration of the switch, and it is loaded when the switch is first installed and powered on.

The **load factory default** command is a standard Junos OS configuration command. This configuration command replaces the current active configuration with the default factory configuration.

To revert the switch to the rescue configuration:

```
[edit]
user@switch# load factory-default
[edit]
user@switch# delete system commit factory-settings
[edit]
user@switch# commit
```

**Related
Documentation**

- [Understanding Configuration Files](#)
- [Loading a Previous Configuration File on page 183](#)
- [Reverting to the Rescue Configuration on page 184](#)

Reverting to the Rescue Configuration

If someone inadvertently commits a configuration that denies management access to a device and the console port is not accessible, you can overwrite the invalid configuration and replace it with the rescue configuration. The rescue configuration is a previously committed, valid configuration.

To revert the switch to the rescue configuration:

1. Enter the **load override** command.

```
[edit]
user@switch# load override filename
```

2. Commit your changes.

```
[edit]
user@switch# commit filename
```


Related Documentation • [Reverting to the Default Factory Configuration on page 184](#)

Recovering the Root Password

If you forget the root password, you can use the password recovery procedure to reset the root password.



NOTE: The root password cannot be recovered on a QFabric system.



NOTE: You need console access to the switch to recover the root password.

To recover the root password:

1. Power off the switch by switching off the AC power outlet of the device or, if necessary, by pulling the power cords out of the device's power supplies.
2. Turn off the power to the management device, such as a PC or laptop computer, that you want to use to access the CLI.
3. Plug one end of the Ethernet rollover cable supplied with the device into the RJ-45-to-DB-9 serial port adapter supplied with the device.
4. Plug the RJ-45-to-DB-9 serial port adapter into the serial port on the management device.
5. Connect the other end of the Ethernet rollover cable to the console port on the device.
6. Turn on the power to the management device.
7. On the management device, start your asynchronous terminal emulation application (such as Microsoft Windows Hyperterminal) and select the appropriate **COM** port to use (for example, **COM1**).
8. Configure the port settings as follows:
 - Bits per second: 9600
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: None
9. Power on the device by (if necessary) plugging the power cords into the device's power supply, or turning on the power to the device by switching on the AC power outlet the device is plugged into.

The terminal emulation screen on your management device displays the device's boot sequence.

10. When the following prompt appears, press the Spacebar to access the device's bootstrap loader command prompt:

```
Hit [Enter] to boot immediately, or space bar for command prompt.  
Booting [kernel] in 9 seconds...
```

11. At the following prompt, enter **boot -s** to start up the system in single-user mode.

```
ok boot-s
```

12. At the following prompt, enter **recovery** to start the root password recovery procedure.

```
Enter full pathname of shell or 'recovery' for root password recovery or RETURN  
for /bin/sh: recovery
```

13. Enter configuration mode in the CLI.

14. Set the root password. For example:

```
user@switch# set system root-authentication plain-text-password
```

15. At the following prompt, enter the new root password. For example:

```
New password: juniper1  
Retype new password:
```

16. At the second prompt, reenter the new root password.

17. After you have finished configuring the password, commit the configuration.

```
root@host# commit  
commit complete
```

18. Exit configuration mode in the CLI.

19. Exit operational mode in the CLI.

20. At the prompt, enter **y** to reboot the device.

```
Reboot the system? [y/n] y
```

Related Documentation

- *Configuring the Root Password*